

**ФАКУЛЬТЕТ УПРАВЛІННЯ ПРОЦЕСАМИ ПЕРЕВЕЗЕНЬ**

**3.1.1 Кафедра «Управління вантажною і комерційною роботою»**

**СУЧАСНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В УПРАВЛІННІ  
ЗАЛІЗНИЧНИМИ ПІДРОЗДІЛАМИ**

**МЕТОДИЧНІ ВКАЗІВКИ**

**для самостійної роботи студентів спеціальності 070101 «Організація перевезень та управління на залізничному транспорті» всіх форм навчання та слухачів ІПКК**

**Харків 2010**

Методичні вказівки розглянуто та рекомендовано до друку на засіданні кафедри “Управління вантажною і комерційною роботою” 29 вересня 2008 р., протокол № 2.

У методичних вказівках наведені загальні вимоги щодо формувань поняття сучасних інформаційно-керуючих технологій та знань програмного забезпечення цього виду діяльності залізниць, навички користування сучасними методиками і програмним забезпеченням.

Призначено для самостійної роботи студентів спеціальності ОПУТ всіх форм навчання та слухачів ІППК.

Укладачі:

проф. Д.В. Ломотько

доц. Д.В. Шумик

асист. А.Л. Обухова

Рецензент

доц. О.М. Огар

## СУЧАСНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В УПРАВЛІННІ ЗАЛІЗНИЧНИМИ ПІДРОЗДІЛАМИ

### МЕТОДИЧНІ ВКАЗІВКИ

для самостійної роботи студентів спеціальності 070101 «Організація перевезень та управління на залізничному транспорті» всіх форм навчання та слухачів ІППК

Відповідальний за випуск Шумик Д.В.

Редактор Губарева К.А.

---

Підписано до друку 27.11.08 р.

Формат паперу 60x84 1/16 . Папір писальний.

Умовн.-друк.арк. 2,5 Обл.-вид.арк. 2,75.

Замовлення № Тираж 300. Ціна

---

Видавництво УкрДАЗТу, свідоцтво ДК 2874 від 12.06.2007 р.

Друкарня УкрДАЗТу,  
61050, Харків - 50, майд. Фейсрбаха, 7

## ЗМІСТ

Перелік скорочень.....	5
1 Сучасні методики представлення та оброблення інформації. База даних.....	6
1.1 Поняття баз даних, у тому числі реляційних. СУБД, що використовуються.....	6
1.2 Проектування та нормалізація реляційних баз даних.....	13
1.3 Вступ до мови SQL. Мова SQL як стандартна мова баз даних. Пошук інформації.....	17
1.4 Формування запитів на мові SQL.....	25
1.4.1 Основні елементи мови SQL.....	25
1.4.2 Функціональні елементи команд SQL.....	26
2 Методи оптимізації обсягу інформації, що зберігається.....	29
2.1 Поняття, пов'язані з інформаційними технологіями в галузі перевезень.....	29
2.2 Безпека та захист даних.....	30
2.3 Особливості застосування інформаційних систем на залізничному транспорті України.....	34
3 Автоматизована система керування пасажирськими перевезеннями на залізничному транспорті України....	36
3.1 Термінальне устаткування.....	36
3.1.1 Пульт касира.....	37
3.2 Нормативно-довідкова інформація.....	38
3.2.1 Загальні положення щодо кодування інформації. Кодування залізниць, інформаційних статистичних центрів, залізничних адміністрацій	38
3.2.2 Кодування поїздів та ниток.....	41
3.2.3 Кодування вагонів.....	41
3.2.4 Кодування станцій.....	42
3.2.5 Регулювання продажу місць у поїздах.....	43
3.3 Квитково-касові операції.....	45
3.3.1 Загальні положення.....	45
4 Автоматизована система керування вантажними перевезеннями на залізничному транспорті України....	46
4.1 Основні відомості.....	47

4.2 Технологія функціонування.....	48
4.2.1 Функції серверної частини.....	48
4.2.2 Функції клієнтської частини.....	49
4.2.3 Методика взаємодії АРМ ТРА та АРМ РНЗ	50
Контрольні питання з дисципліни.....	51
Список літератури.....	52

## Перелік скорочень

- АРМ РНЗ – автоматизоване робоче місце з розрахунку норм закріплення
- АРМ ТВК – автоматизоване робоче місце товарного касира
- АРМ ТРА – автоматизоване робоче місце інженера з ТРА
- АСК ВП УЗ – автоматизована система керування вантажними перевезеннями на залізничному транспорті України
- АСК ПП УЗ – автоматизована система керування пасажирськими перевезеннями на залізничному транспорті України
- АСУ – автоматизована система управління
- БД – база даних
- ГІОЦ – головний інформаційно-обчислювальний центр
- ЕОМ – електронна обчислювальна машина
- ІСЦ – інформаційний статистичний центр
- КПД – комплект перевізних документів
- НДІ – нормативно-довідкова інформація
- ОДБ – об'єднане дорожнє бюро
- ОК – обчислювальний комплекс
- СУБД – системи управління базами даних
- ТІ – теорія інформації
- ТРА – техніко-розпорядчий акт

## **1 Сучасні методики представлення та оброблення інформації. База даних**

Основою інформаційного забезпечення автоматизованої системи управління (АСУ) є бази даних (БД) та бази знань. База даних являє собою сукупність даних та складається з багатьох таблиць. Окремий рядок у таблиці БД називається *записом*. Кожний запис представляє інформацію про окремий об'єкт. Наприклад, запис про вагон може містити таку інформацію: номер вагона, тип вагона, адресу приписки та таке інше. Базою даних, наприклад, є телефонна книга. Інформація, що в ній міститься, складається з прізвищ, адрес та телефонних номерів. Дана інформація розміщена в алфавітній послідовності. На мові баз даних це означає, що вона проіндексована та дозволяє легко і швидко знайти необхідного абонента.

Будь-яка база даних потребує супроводу. У телефонній книзі можливі доповнення нових номерів, корегування адрес вже раніше вказаних людей. Для здійснення цієї мети використовуються спеціальні системні засоби, які отримали назву системи управління базами даних (СУБД).

### **1.1 Поняття баз даних, у тому числі реляційних. СУБД, що використовуються**

СУБД – програма, що управляє роботою бази даних та дозволяє записувати інформацію у базу, знайти необхідний запис, прочитати його, а також контролювати цілісність бази даних. Більшість існуючих СУБД може бути поділено на такі класи:

- за кількістю одночасно працюючих з базою даних користувачів СУБД можна поділити на *використовні* та *багатовикористовні*;
- за структурою організації даних СУБД можна поділити на реляційні та мережні бази даних.

На сьогодні основну роль відіграють реляційні бази даних. **Реляційною** називається база даних, яка поділена на логічно пов'язані між собою складові, що іменуються таблицями. У реляційній базі даних інформація розбита на невеликі, логічно пов'язані й тому більш керувані елементи, які через рівень своєї організації спрощують її супровід та забезпечують найбільш оптимальне функціонування.

Таблиця – спосіб зберігання інформації у БД. Кожна таблиця має унікальне поле, що характеризує конкретний запис у таблиці – ключ.

На залізничному транспорті найбільш розповсюдженими є такі СУБД:

1 **MS SQL Server 2000**. Розробка фірми Microsoft. Існують тільки під управлінням операційної системи Windows NT/Windows 2000. Найбільш розповсюджений варіант для не зовсім великих додатків.

2 **Oracle**. Продукт фірми Oracle Technology Corporation. Існують версії як для платформи Windows, так і для платформи Unix. Найбільш розвинута, важка, ресурсомістка з нині існуючих СУБД. Використовується, як правило, тільки для великих додатків.

3 **DB2**. Продукт фірми IBM. Використовується на великих машинах для завдань, що обслуговують значну кількість користувачів. Розробка додатків на DB2 досить трудомістка. Система у порівнянні з двома попередніми значно менш розповсюджена.

Вибір СУБД є відповідальним та найважливішим моментом на етапі проектування корпоративної мережі. Найбільш розповсюдженим на сьогодні СУБД є Oracle 8i та MS SQL Server 2000. При обиранні однієї з них враховуються такі фактори: обсяг інформації, що зберігається; вартість СУБД; наявність висококваліфікованих спеціалістів (адміністратор БД, проектувальник БД, розробник сервера додатків та інш.). MS SQL Server 2000 – один з найбільш розповсюджених і простих в управлінні та супроводженні. З точки зору перелічених вище факторів та перспективного розвитку мережі, доцільним буде обирати Oracle 8i Server. Сервер

Oracle 8 дозволяє зберігати дуже великі масиви даних та надавати користувачам швидкий доступ до цих даних. Сервер Oracle 8 забезпечує сумісне використання інформації, що зберігається, багатьма одночасно працюючими додатками. При цьому інформація зберігається в одному місці, але використовується великою кількістю різних систем. Сервер Oracle 8 містить цілий ряд переваг, які дозволяють йому бути лідером на ринку засобів управління інформацією. Назвемо головні.

**Безпека та захист даних** від несанкціонованого доступу. Кожний користувач отримує обмежені права на перегляд, модифікацію та створення суворо відповідних видів даних.

**Резервне копіювання та відновлення даних.** Мінімізується як можливість втрачання даних, так і час відновлення працездатності системи після збою. При цьому гарантується цілодобова доступність бази даних.

**Управління дисковим простором.** Це дозволяє резервувати відповідну частину дискового простору для майбутнього використання та контролю за розподілом вільного простору.

**Відкриті стандарти міжсистемної взаємодії** забезпечують взаємодію з програмними продуктами інших фірм. Використовуючи додаткові додатки Oracle, користувачі мають можливість роботи з даними, що накопичені в базі даних DB2, Sybase або Microsoft Access.

**Засоби розробки додатків** надають широкий спектр засобів, включаючи засоби генерації запитів кінцевими користувачами, готові прикладні системи, а також засоби управління інформацією у масштабі офісу.

**Засоби забезпечення цілісності даних** дозволяють відмінити всі незавершені транзакції у випадку будь-якого збою при модифікації даних різними користувачами.

**Наявність процедурних компонент,** які є розширеною мовою PL/SQL. Процедурні компоненти дозволяють створювати збережувані процедури, тригери баз даних, пакети збережуваних процедур (зберігаються у базі даних у вигляді єдиної програмної одиниці). Процедурні



компоненти сервера Oracle забезпечують оброблення запитів на мові PL/SQL. Ці програмні компоненти є доступними для всіх додатків.

**Засоби роботи з відеоданими (Video Option)** суттєво розширюють можливості сервера Oracle 8 із зберігання, управління та відтворення у реальному часі повноекранного відео та високочастотного звуку. Подібні системи дозволяють автоматично відстежувати оперативну обстановку на станціях дороги, користуючись оперативною інформацією у базі даних Oracle.

Для того щоб прийняти рішення про вибір СУБД, необхідно розглянути основні характеристики СУБД MS SQL Server 2000.

- **Швидкодія.** MS SQL Server 2000 досить швидкодіюча СУБД. На думку розробників, СУБД MS SQL Server 2000 є однією з найшвидших баз даних серед всіх існуючих на цей час на ринку.

- **Простота використання.** СУБД MS SQL Server 2000 є високопродуктивною та відносно простою у використанні. Вона значно легше інсталюється.

- **Ціна.** СУБД MS SQL Server 2000 розповсюджується безкоштовно для домашнього використання, а для організацій ціна визначається кількістю користувачів мережі.

- **Підтримка мови запитів.** MS SQL Server 2000 «розуміє» команди мови SQL. Ця мова застосовується у всіх сучасних СУБД. MS SQL Server 2000 також підтримує інтерфейс ODBC (Open Database Connectivity), протокол інтерфейсу з базами даних, розроблений компанією Microsoft. Підтримка ODBC дуже важлива, бо при наявності будь-яких платформ та систем, що використовуються в обчислювальній мережі, виникає необхідність обміну даними між різними СУБД.

- **Можливості.** Сервер дозволяє одночасне підключення необмеженій кількості користувачів. Доступ до СУБД MS SQL Server 2000 можливо здійснити в інтерактивному режимі за допомогою різних інтерфейсів, що дозволяють вводити запити та переглядати отримані результати.

- **Взаємодія та безпека.** СУБД MS SQL Server призначена для роботи в мережі та має можливість доступу через Інтернет, тобто можливість роботи з даними в будь-якій точці. При цьому СУБД MS SQL Server споряджена розвинутою системою захисту від несанкціонованого доступу.

- **Переносність.** СУБД MS SQL Server може працювати як під управлінням будь-яких версій UNIX, так і під управлінням систем, що не використовують UNIX, таких, як Windows та OS/2.

- **Відкрите розповсюдження.** Дистрибуція СУБД MS SQL Server 2000 є легкодоступною. Для цього досить скористатися веб-сервером.

- **Динамічне резервне копіювання, автоматичне поновлення** бази даних у випадку збою обладнання, відключення живлення без втручання адміністратора. Всі завершені, але не відображені в базі даних транзакції з журналу транзакцій, застосовувані до бази даних. Незавершені транзакції вилучаються з журналу.

- **Система привілеїв.** СУБД MS SQL Server 2000 забезпечує користувачам можливість виконувати тільки ті дії, які їм дозволені відповідно до їх обов'язків. Коли користувач підключається до сервера MS SQL, його особа встановлюється за іменем хоста, з якого він підключився, та іменем користувача. Система надає привілеї відповідно до особистих даних та тих, що ви бажаєте робити. При підключенні до MS SQL сервер перевіряє, чи взагалі має користувач дозвіл на підключення. Якщо так, то сервер перевіряє кожний його запит для того, щоб переконатися, що користувач має досить привілеїв для його виконання.

Необхідно зазначити, що СУБД Oracle 8i за переліченими параметрами не поступається СУБД MS SQL Server, а за деякими параметрами є лідером, а саме, за масштабуванням, надійністю, безпекою, управлінням даними, об'єктною орієнтацією.

**Розподілені бази даних.** Всі СУБД призначені для зберігання та оброблення інформації. Реляційний підхід до управління базами даних заснований на математичній

моделі, що використовує методи реляційної алгебри та реляційного обчислення.

Реляційна СУБД повинна:

- надавати всю інформацію у вигляді таблиць;
- підтримувати логічну структуру даних незалежно від їх фізичного подання;
- використовувати мову високого рівня для структурування, виконання запитів та зміни інформації у базі даних (для цього використовується мова SQL);
- підтримувати основні операції (обирання, проектування, перетин та доповнення);
- підтримувати віртуальні таблиці, забезпечуючи користувачам альтернативний спосіб перегляду даних у таблицях;
- розрізняти в таблицях невідомі значення, нульові значення та пропуски даних;
- забезпечувати механізм для підтримки цілісності, авторизації, транзакції та відновлення даних.

При розробленні автоматизованої системи дуже важливим є правильно обрати структуру бази даних та правильно визначити стратегію розподілу даних. Організація розподілу баз даних була підготовлена розширенням можливостей персональних комп'ютерів та більшості ЕОМ, а також розвитком мережних засобів їх взаємодії.

При організації БД можливі три альтернативних стратегії розподілу даних:

а) централізована, коли одна єдина копія бази даних розміщується на одній обчислювальній машині, наприклад в одному ПЛОЦ;

б) розподілена, коли бази даних розподілені між декількома, зазвичай розміщеними у різних місцях, обчислювальними машинами, між якими організований інформаційний простір. Таким чином, організують неперетинні підмножини однієї єдиної копії бази даних, наприклад, бази даних, розподілені за різними ІОЦ;

в) змішана, коли у кожному ІОЦ може міститися фрагмент бази даних (за власною дорогою), а в ПІОЦ Укрзалізниці буде повна копія бази даних.

Всі розглянуті підходи є досить ефективними, однак існують як переваги, так і недоліки кожної з трьох стратегій, тому остаточний вибір стратегії розподілу даних може бути зроблений тільки після детального аналізу. Це здійснюється на етапі проектування системи.

**Локальні бази даних.** При роботі з локальними базами даних сама БД розміщена на тому ж самому комп'ютері, що і додатки, які здійснюють доступ до них. Робота з БД виконується у режимі використання одним користувачем. Кожен з користувачів має на своєму комп'ютері локальну копію даних. Копія час від часу обновлюється з реальної БД, що розміщена на мережному сервері. Відмінностей, з точки зору архітектури, між (одновикористовною) архітектурою та архітектурою «файл-сервер» немає. І в тому і в іншому випадках як СУБД застосовуються так звані *персональні* (або *локальні*) СУБД, такі, як Paradox, dBase і т.д. Сама БД у цьому випадку представляє собою набір таблиць, індексних файлів та т. ін., що зберігаються в одному каталозі на диску у вигляді окремих файлів.

**Віддалені бази даних.** В архітектурі «клієнт-сервер» функції додатка *клієнта* та *сервера* розділені. Додаток-клієнт на структурній мові SQL формує запит до сервера, на якому розміщена БД. Віддалений сервер приймає запит та переадресовує його SQL-серверу БД. SQL-сервер – це спеціальна програма, яка управляє віддаленою базою даних. SQL-сервер забезпечує інтерпретацію запиту, його виконання у базі даних, формування результату виконання запиту та видачу його додатку-клієнту. У даному випадку по мережі «подорожують» тільки ті дані, які є необхідними користувачеві. Як результат – знижується навантаження на мережу. В архітектурі «клієнт-сервер» використовуються так звані віддалені (або промислові) СУБД. Вони можуть забезпечити роботу інформаційних систем у масштабі великого підприємства, організації, дороги.

## 1.2 Проектування та нормалізація реляційних баз даних

У процесі проектування (конструювання) вирішується, який вигляд буде мати знов створювана база даних; при цьому необхідно обрати:

- а) таблиці, які будуть входити до бази даних;
- б) стовпчики, з яких складатиметься кожна таблиця;
- в) взаємозв'язки між таблицями та стовпчиками.

Конструювання бази даних пов'язано з побудовою її логічної структури. У реляційній моделі логічна структура бази абсолютно не залежить від її фізичної структури та способу зберігання. Логічна структура також не визначається тим, що бачить на своєму екрані кінцевий користувач. Наприклад, це можуть бути віртуальні таблиці, створені розробником або прикладними програмами.

Конструювання бази даних на основі реляційних моделей має ряд важливих переваг перед іншими моделями, а саме:

- незалежність логічної структури від фізичної уяви та уяви користувача;
- гнучкість структури БД – конструктивні рішення не обмежують можливості виконання у майбутньому будь-яких різноманітних запитів;
- реляційна модель не потребує опису всіх можливих зв'язків між даними.

З часом можливо задавати запити про будь-які логічні взаємозв'язки, що містяться у базі, а не тільки про ті, які первісно планувалися.

З іншого боку, реляційні системи не мають жодних вбудованих захисних механізмів проти некоректних структурних рішень та не вміють розрізняти добру структуру бази даних від абиякої. До того ж не існує автоматизованих засобів, які могли б замінити розробника або користувача у процесі прийняття структурних рішень.

Нижче наведено зразкову послідовність кроків, що виконується у процесі проектування бази даних.

I Дослідження інформаційного середовища для моделювання передбачає відповіді на такі питання:

1 Звідки надходить інформація та в якому вигляді?  
2 Яким чином вона буде вводиться у систему та хто цим буде займатися?

3 Як часто змінюється інформація?

4 Які параметри системи будуть найбільш критичними: час та реакція на запит або надійність?

З цією метою є потреба в:

- вивченні всіх паперових матеріалів, а також інформаційних файлів та форм, які використовуються в організації для зберігання та оброблення даних;

- уточненні, у якому вигляді інформація повинна бути вилучена з бази даних – у формі звіту, замовлень, статистичної інформації та інше;

- зазначені, кому буде призначена вихідна інформація.

II Створення переліку об'єктів (сутностей, які будуть предметом бази даних) разом з їх властивостями та атрибутами. Об'єкти повинні бути складені у вигляді таблиць (кожний рядок дає опис одного об'єкта, наприклад: вагон, станція, локомотив). Властивості об'єктів будуть представлені стовпчиками таблиці, наприклад: номер, тип, приписка та т. ін. (рисунок 1.1).

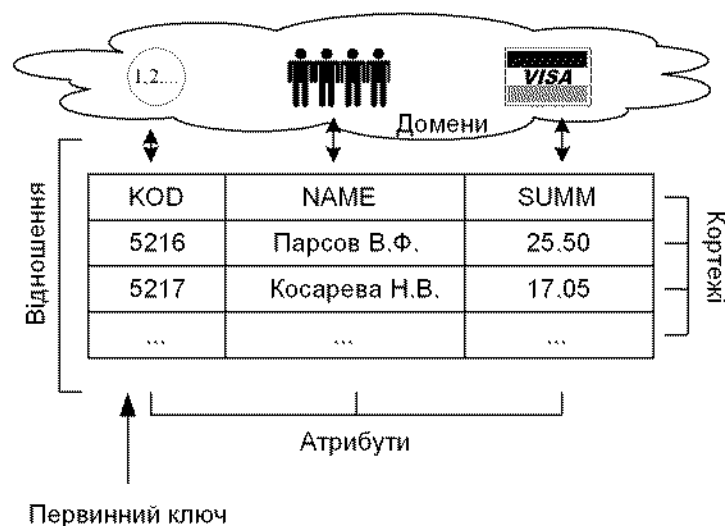


Рисунок 1.1 – Основні поняття реляційних баз даних

III У ході роботи є обов'язковим створення макета таблиць та зв'язків між ними, що має назву структури даних або діаграми залежностей між об'єктами (ER-діаграма).

IV Попередньо розібравшись з об'єктами та їх атрибутами, необхідно переконатися, що кожний об'єкт має атрибут (або групу атрибутів), відповідно до якого однозначно можливо ідентифікувати будь-який рядок у майбутній таблиці. Цей ідентифікатор зазвичай називається **первинним ключем**. Якщо такого немає, то для отримання штучного ключа слід створити додатковий стовпчик.

V Повинні бути розглянуті залежності між об'єктами (рисунок 1.2):

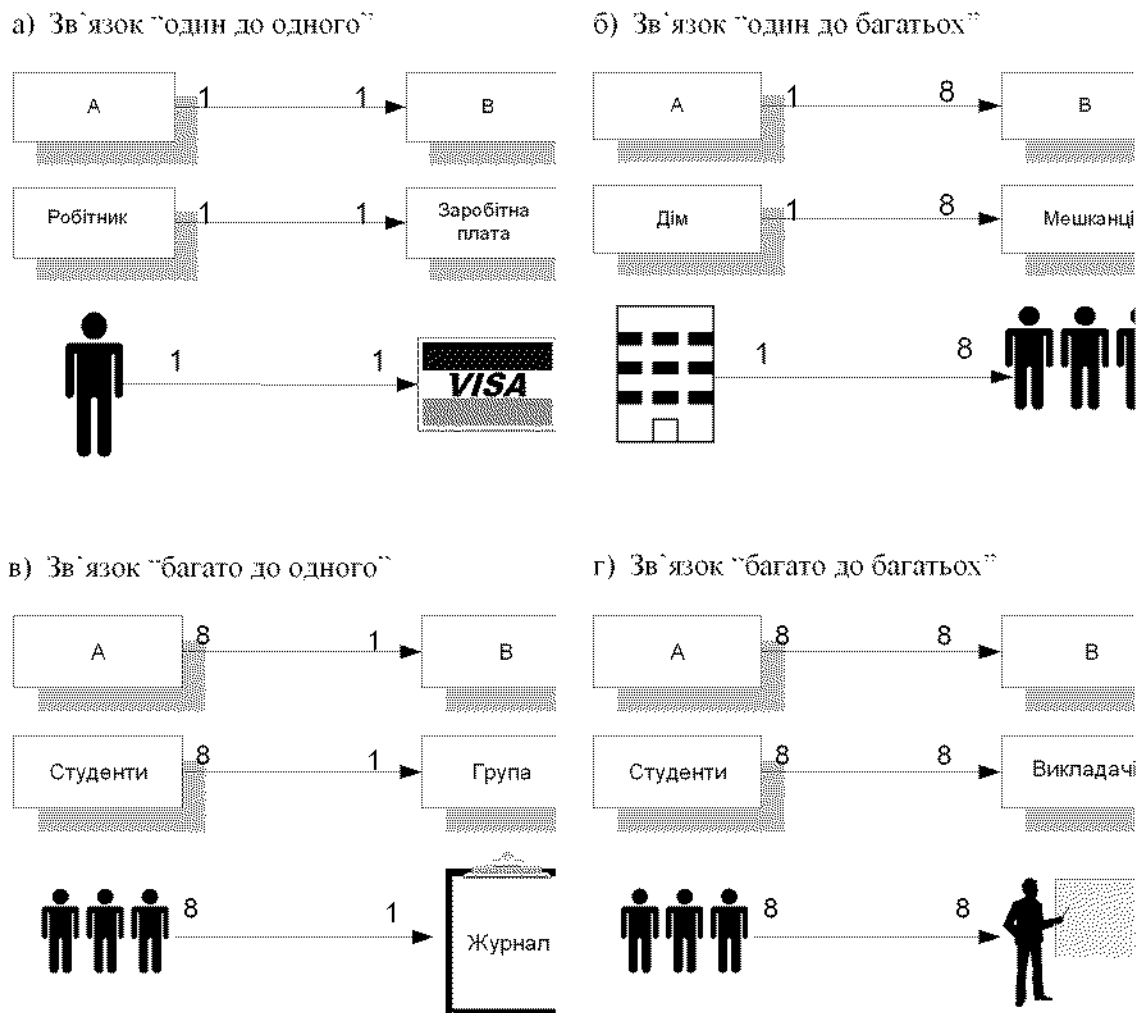


Рисунок 1.2 – Залежності між об'єктами. Зв'язки

а) чи є у наявності залежність типу «один до багатьох» (наприклад, одна бригада може обслуговувати багато локомотивів, але кожний локомотив може бути обслугований однією бригадою) або «багато до багатьох»?

б) чи є можливості для об'єднання пов'язаних таблиць? Для цього призначені зовнішні ключі, стовпчики у пов'язаних таблицях із співпадаючими значеннями первинних ключів.

VI Аналіз структури бази даних з точки зору правил нормалізації для пошуку логічних помилок. Виправлення всіх відхилень від нормальних форм або обґрунтування рішення відмовитися від виконання ряду правил нормалізації в інтересах простоти або продуктивності. Документування причин таких рішень.

VII Безпосереднє створення структури бази даних та розміщення в ній деяких прототипів даних. Обов'язкове експериментування із запитами, вивчення отриманих результатів. Виконання ряду текстів на продуктивність для того, щоб перевірити різні технічні рішення.

VIII Оцінка бази даних з точки зору того, чи задовольняють замовника отримані результати.

Необхідно відмітити, що широкі таблиці важко читати, а також в них важко розібратися. У той же час поділ даних таблиць на невеликі таблиці ускладнює відстеження взаємозв'язків між ними. Вибір відповідної кількості стовпчиків зазвичай є компромісом між простотою розуміння бази та правилами нормалізації.

Добре розроблена база даних запобігає введенню суперечливої інформації та випадковому видаленню даних. Це досягається за рахунок мінімізації непотрібного дублювання даних у таблицях та підтримки цілісності. Добре розроблена база даних повинна володіти достатньою продуктивністю. Знов таки тут велику роль відіграє кількість стовпчиків у таблиці. Вибірка даних буде проводитися повільніше, якщо інформація розміщена не в одній, а в декількох таблицях. Однак великі таблиці можуть вимагати від системи оброблення більшої кількості даних, ніж це насправді необхідно для виконання конкретного запиту. Іншими словами, кількість та розмір таблиць суттєво



впливають на продуктивність. З точки зору продуктивності критичним є вибір стовпчика, з якого виконується індексування, та тип індексування.

Погана структура бази даних:

- призводить до нерозуміння результатів виконання запитів;

- підвищує ризик введення до бази даних суперечливої інформації;

- породжує надлишкові дані;

- ускладнює виконання змін структури створених раніше та вже заповнених даних таблиць.

Не існує ідеального рішення, яке повністю б задовольняло всі вимоги, що висуваються при проектуванні баз даних. Часто приходиться від чогось відмовлятися, спираючись на вимоги та особливості додатків, які будуть використовувати базу даних.

### **1.3 Вступ до мови SQL. Мова SQL як стандартна мова баз даних. Пошук інформації**

Стрімке зростання популярності SQL є однією з найважливіших тенденцій у сучасній комп'ютерній промисловості. За останні роки SQL стала єдиною мовою баз даних. На сьогоднішній день SQL підтримують понад сто СУБД, що працюють як на персональних комп'ютерах, так і на великих ЕОМ. Був прийнятий, а потім доповнений офіційний міжнародний стандарт на SQL. Мова SQL є важливою ланкою в архітектурі систем керування базами даних, що випускаються всіма провідними постачальниками програмних продуктів, і служить стратегічним напрямком розробок компанії Microsoft у галузі баз даних. Зародившись у результаті виконання другорядного дослідного проекту компанії IBM, SQL сьогодні широко відома і як потужний ринковий фактор.

**Мова SQL.** SQL є інструментом, призначеним для оброблення і читання даних, що утримуються у комп'ютерній базі даних. SQL – це скорочена назва структурованої мови

запитів (Structured Query Language). SQL є мовою програмування, що застосовується для організації взаємодії користувача з базою даних. Насправді SQL працює тільки з базами даних одного певного типу, названих *реляційними*. На рисунку 1.3 зображена схема роботи SQL. Відповідно до цієї схеми в обчислювальній системі є база даних, у якій зберігається важлива інформація. Якщо обчислювальна система ставиться до сфери бізнесу, то в базі даних може зберігатися інформація про матеріальні цінності, обсяги продажів і зарплату.

У базі даних на персональному комп'ютері може зберігатися інформація про виписані чеки, телефони й адреси або інформація, витягнута з більшої обчислювальної системи. Комп'ютерна програма, що управляє базою даних, називається *системою управління базою даних*, або СУБД.

Якщо користувачеві необхідно прочитати дані з бази даних, він запитує їх у СУБД за допомогою SQL. СУБД обробляє запит, знаходить необхідні дані й посилає їхньому користувачеві. Процес запиту даних і одержання результату називається *запитом до бази даних*, звідси й назва – *структурована мова запитів*.

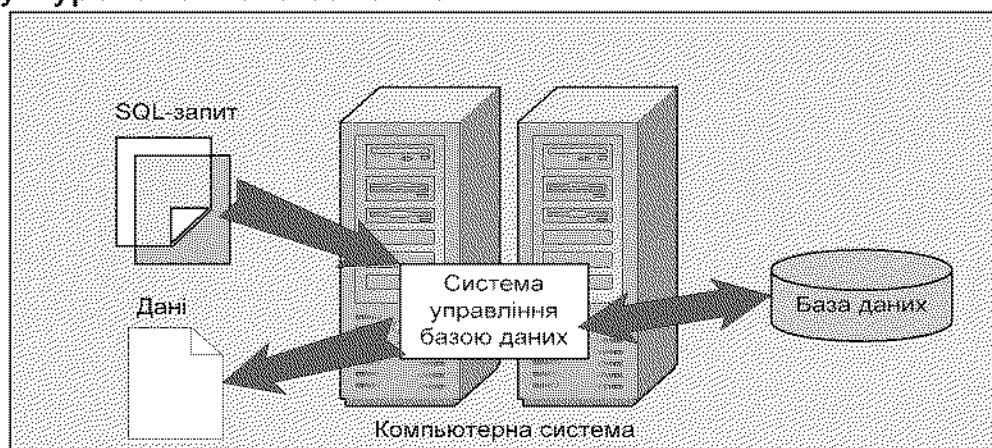


Рисунок 1.3 – Схема роботи SQL – запиту

Однак ця назва не зовсім відповідає дійсності. По-перше, сьогодні SQL являє собою щось набагато більше, ніж простий інструмент створення запитів, хоча саме для цього вона і була спочатку призначена. Незважаючи на те, що читання даних, як і раніше, залишається однією з найбільш важливих функцій

SQL, зараз ця мова використовується для реалізації всіх функціональних можливостей, які СУБД надає користувачеві, а саме:

- *організації даних.* SQL дає користувачеві можливість змінювати структуру подання даних, а також встановлювати відношення між елементами бази даних.

- *читання даних.* SQL дає користувачеві або додатку можливість читати з бази даних дані, що містяться в ній, і користуватися ними.

- *оброблення даних.* SQL дає користувачеві або додатку можливість змінювати базу даних, тобто додавати до неї нові дані, а також видаляти або обновляти вже наявні в ній дані.

- *керування доступом.* За допомогою SQL можна обмежити можливості користувача з читання та зміни даних і захистити їх від несанкціонованого доступу.

- *спільного використання даних.* SQL координує спільне використання даних користувачами, що працюють паралельно, щоб вони не заважали один одному.

- *цілісності даних.* SQL дозволяє забезпечити цілісність бази даних, захищаючи її від руйнування через неузгоджені зміни або відмову системи. Таким чином, SQL є досить потужною мовою для взаємодії із СУБД.

По-друге, SQL - це не повноцінна комп'ютерна мова типу COBO, FORTRAN або C. У SQL немає оператора IF для перевірки умов, немає оператора GOTO для організації переходів і немає операторів DO або FOR для створення циклів. SQL є *підмовою* баз даних, до яких входить біля тридцяти операторів, призначених для керування базами даних. Оператори SQL *вбудовуються* у базову мову, наприклад COBO, FORTRAN або C, і дають можливість одержувати доступ до баз даних. Крім того, з такої мови, як C, оператори SQL можуть посилати СУБД у явному вигляді, використовуючи *інтерфейс викликів функцій*.

Нарешті, SQL - це слабо структурована мова, особливо в порівнянні з такими сильно структурованими мовами, як C або Pascal.

**Достоїнства SQL.** SQL – це легка для розуміння мова й у той же час універсальний програмний засіб керування даними.

Успіх мові SQL принесли такі її особливості:

- незалежність від конкретних СУБД;
- переносимість з однієї обчислювальної системи на іншу;
- наявність стандартів;
- схвалення компанією IBM (СУБД DB2);
- підтримка з боку компанії Microsoft (протокол ODBC);
- реляційна основа;
- високорівнева структура, що нагадує англійську мову;
- можливість виконання спеціальних інтерактивних запитів;
- забезпечення програмного доступу до баз даних;
- можливість різного подання даних;
- повноцінність як мови, призначеної для роботи з базами даних;
- можливість динамічного визначення даних;
- підтримка архітектури клієнт/сервер.

Всі перераховані вище фактори спричинили те, що SQL стала стандартним інструментом для керування даними на персональних комп'ютерах, міні-комп'ютерах і великих ЕОМ. Нижче ці фактори розглянуті більш докладно.

- *SQL – мова адміністрування баз даних.* Адміністратор бази даних, що перебуває на міні-комп'ютері або на великому ЕОМ, використовує SQL для визначення структури бази даних і керування доступом до даних.

- *SQL – мова створення додатків клієнт/сервер.* У програмах для персональних комп'ютерів SQL використовується для організації зв'язку через локальну мережу із сервером бази даних, у якій зберігаються спільно використовувані дані. У більшості нових додатків використовується архітектура клієнт/сервер, що дозволяє звести до мінімуму мережний трафік і підвищити швидкодію як персональних комп'ютерів, так і серверів баз даних.

- *SQL – мова розподілених баз даних.* У системах керування розподіленими базами даних SQL допомагає розподіляти дані серед декількох взаємодіючих обчислювальних систем. Програмне забезпечення кожної системи за допомогою використання SQL зв'язується з іншими системами, посилаючи їм запити на доступ до даних.

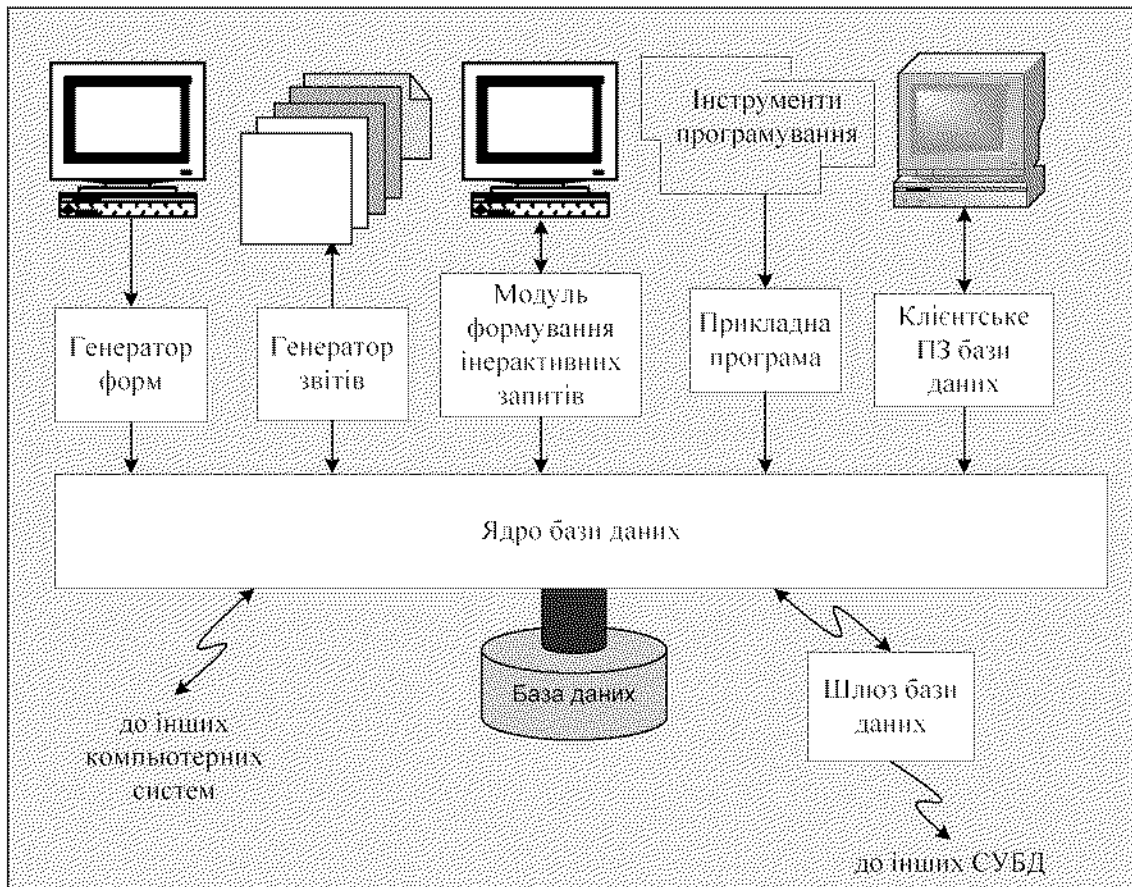


Рисунок 1.4 – Структурна схема типової СУБД

Незважаючи на не зовсім точну назву, SQL на сьогоднішній день є *єдиною* стандартною мовою для роботи з реляційними базами даних. SQL – це досить потужна і в той же час відносно легка для вивчення мова.

**Роль SQL.** Сама по собі мова SQL не є ні системою керування базами даних, ні окремим програмним продуктом. Не можна піти в комп'ютерний магазин і “купити SQL”. SQL – це невід'ємна частина СУБД, інструмент, за допомогою якого здійснюється зв'язок користувача з нею. На рисунку 1.4 наведена структурна схема типової СУБД, компоненти якої з'єднуються в єдине ціле за допомогою SQL.

*Ядро бази даних* є серцевиною СУБД; воно відповідає за фізичне структурування і запис даних на диск, а також за фізичне читання даних з диска. Крім того, воно приймає SQL-запити від інших компонентів СУБД (таких, як генератор форм,

генератор звітів або модуль формування інтерактивних запитів), від користувальницьких додатків і навіть від інших обчислювальних систем. Як видно з рисунка, SQL виконує багато різних функцій:

- SQL – *інтерактивна мова запитів*. Користувачі вводять команди SQL в інтерактивні програми, призначені для читання даних і відображення їх на екрані. Це зручний спосіб виконання спеціальних запитів.

- SQL – *мова програмування баз даних*. Щоб одержати доступ до бази даних, програмісти вставляють у свої програми команди SQL. Ця методика використовується як у програмах, написаних користувачами, так і в службових програмах баз даних (таких, як генератори звітів і інструменти введення даних).

**Стандарти мови.** Офіційний стандарт мови SQL був опублікований Американським інститутом національних стандартів (American National Standards Institute - ANSI) і Міжнародною організацією зі стандартів (International Standards Organization - ISO) у 1986 році та значно розширений у 1992 році. Крім того, ISO є федеральним стандартом США з оброблення інформації (FIPS - Federal Information Processing Standard), а отже, відповідність йому є однією з основних вимог, що утримуються у великих урядових контрактах, які ставляться до галузі обчислювальної техніки. SQL Access Group - консорціум постачальників комп'ютерного встаткування і баз даних - визначив для SQL стандартний інтерфейс викликів функцій, що є основою протоколу ODBC компанії Microsoft і входить також у стандарт X/OPEN. Ці стандарти служать якби офіційною печаткою, що схвалює SQL, і вони прискорили завоювання нею ринку.

**Реляційна основа.** SQL є мовою реляційних баз даних, тому вона стала популярною тоді, коли популярною стала реляційна модель подання даних. Таблична структура реляційної бази даних інтуїтивно зрозуміла користувачам, тому мова SQL є простою і легкою для вивчення. Реляційна модель має солідний теоретичний фундамент, на якому були засновані еволюція і реалізація реляційних баз даних.

На хвилі популярності, викликаній успіхом реляційної моделі, SQL стала *єдиною* мовою для реляційних баз даних.

**Багаторівнева структура, що нагадує англійську мову.** Оператори SQL виглядають як звичайні англійські пропозиції, що спрощує їхнє вивчення і розуміння. Частково це обумовлено тим, що оператори SQL *описують* дані, які необхідно одержати, а не *визначають* спосіб їхнього пошуку. Таблиці та стовпці в реляційній базі даних можуть мати довгі описові імена. У результаті більшість операторів SQL означають саме те, що точно відповідає їхнім іменам, тому їх можна читати як прості, зрозумілі пропозиції.

**Інтерактивні запити.** SQL є мовою інтерактивних запитів, що забезпечує користувачам негайний доступ до даних. За допомогою SQL користувач може в інтерактивному режимі одержати відповіді на найскладніші запити за лічені хвилини або секунди, тоді як програмісту потрібні були б дні або тижні, щоб написати для користувача відповідну програму. Через те що SQL допускає негайні запити, дані стають більш доступними та можуть допомогти в прийнятті рішень, роблячи їх більш обґрунтованими.

**Програмний доступ до бази даних.** Програмісти користуються мовою SQL, щоб писати додатки, у яких містяться звертання до баз даних. Ті самі оператори SQL використовуються як для інтерактивного, так і для програмного доступу, тому частини програм, що містять звертання до бази даних, можна спочатку тестувати в інтерактивному режимі, а потім вбудовувати в програму. У традиційних базах даних для програмного доступу використовуються одні програмні засоби, а для виконання негайних запитів - інші.

**Різні подання даних.** За допомогою SQL творець бази може зробити так, що різні користувачі бази даних будуть бачити *різні* подання її структури і вмісту. Наприклад, базу даних можна спроектувати таким чином, що кожен користувач буде бачити тільки дані, що ставляться до його підрозділу або торговельного регіону. Крім того, дані з різних частин бази даних можуть бути скомбіновані і подані користувачеві у вигляді однієї простої таблиці. Отже,

подання можна використовувати для посилення захисту бази даних і її налаштування під конкретні вимоги окремих користувачів.

**Повноцінна мова для роботи з базами даних.** Спочатку SQL була задумана як мова інтерактивних запитів, але зараз вона вийшла далеко за рамки читання даних. SQL є повноцінною й логічною мовою, призначеною для створення бази даних, управління її захистом, зміни її вмісту, читання даних і спільного використання даних декількома користувачами, що працюють паралельно. Прийоми, освоєні при вивченні одного розділу мови, можуть потім застосовуватися в інших командах, що підвищує продуктивність роботи користувачів.

**Динамічне визначення даних.** За допомогою SQL можна динамічно змінювати та розширювати структуру бази даних навіть у той час, коли користувачі звертаються до її вмісту. Це велика перевага перед мовами статичного визначення даних, які забороняють доступ до бази даних під час зміни її структури. Таким чином, SQL забезпечує максимальну гнучкість, тому що дає базі даних можливість адаптуватися до вимог, що змінюються, не перериваючи роботу додатка, що виконується у реальному масштабі часу.

**Архітектура клієнт/сервер.** SQL – природний засіб для реалізації додатків клієнт/сервер. У цій ролі SQL служить сполучною ланкою між клієнтською системою, взаємодіючою з користувачем, і серверною системою, що управляє базою даних, дозволяючи кожній системі зосередитися на виконанні своїх функцій. Крім того, SQL дозволяє персональним комп'ютерам функціонувати як клієнтам стосовно мережних серверів або великих баз даних, встановлених на великих ЕОМ; це дозволяє отримувати доступ до корпоративних даних з додатків, що працюють на персональних комп'ютерах (рисунок 1.5).



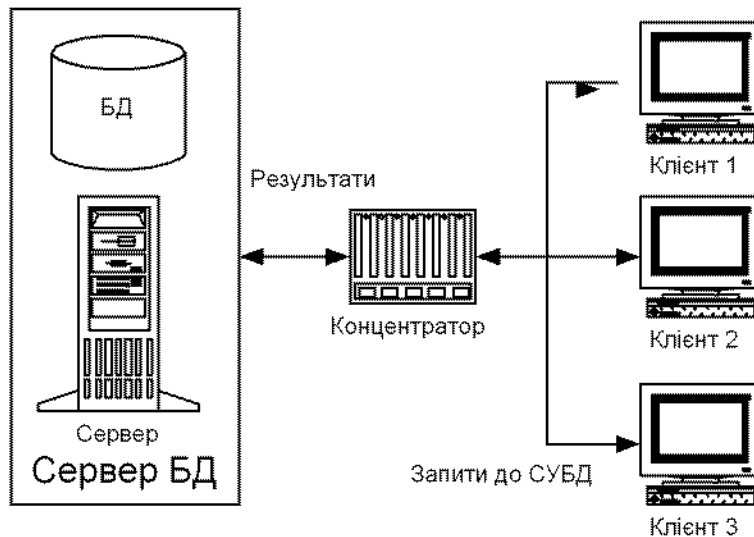


Рисунок 1.5 – Архітектура СУБД клієнт/сервер

## 1.4 Формування запитів на мові SQL

### 1.4.1 Основні елементи мови SQL

- <separator> - роздільник;
- <comment> - коментар, не оброблюваний як команда системою;
- <identifier> - ідентифікатор елемента, відповідно до стандарту ANSI символи повинні бути набрані у верхньому регістрі, а сам ідентифікатор не повинен бути довше 18-ти символів;
- <underscore> - знак "\_";
- <percent sign> - знак "%";
- <string> - друк будь-якого тексту, що поданий в одиночних лапках;
- <delimiter> - кожен з таких знаків: ",", "(", ")", "<", ">", ":", ":", "=", "+", "-", "|", "<>", ">=", "<=", <string>, а також подвійні лапки;
- <SQL term> - закінчення вкладеної команди.

## 1.4.2 Функціональні елементи команд SQL

- <query> - пропозиція SELECT;
- <value expression> <primary> | <operator> <value expression> - конструкція, в якій <operator> може бути кожним з таких: "+", "-", "/", "\*"; <primary> - може бути <column name> - назвою поля таблиці або математичним виразом;
- <table name> - ім'я таблиці даних <identifier>;
- <column spec> - конструкція такої структури [<table name> | <alias>.]<column name>, де <alias> - визначене системою умовне логічне ім'я таблиці даних;
- <colconstraint> NOT NULL | UNIQUE | CHECK (<predicate>) | PRIMARY KEY | REFERENCES <table name>[(<column name>)] - умова перевірки стовпця;
- <defvalue> - значення за замовчуванням, що задається <value expression>;
- <data type> - опис припустимого типу даних;
- <cursor name> - найменування подання даних, визначених за допомогою <identifier>;
- <index name> - найменування певного індексу, визначеного за допомогою <identifier>.

Опис запитів до вагонної моделі приведений в таблиці 1.1.

Елемент бази даних «Вагонна модель станції», наведений за допомогою СУБД Microsoft Access, наведений на рисунку 1.6.

Таблиця 1.1 – Опис запитів до вагонної моделі

Назва запиту	Текст SQL-запиту	Пояснення
Кількість піввагонів за станціями призначення	SELECT COUNT(*) FROM [станція Львів ОКС] WHERE [№ ВАГОНА] 10000000=6	З таблиці станція Львів ОКС (FROM [станція Львів ОКС]) обрати всі записи (SELECT COUNT(*)), де номер вагона починається на 6 (WHERE [№ ВАГОНА] 10000000=6)
Кількість порожніх вагонів	SELECT COUNT(*) FROM [станція Львів ОКС] WHERE [Маса вантажу]=0	З таблиці станція Львів ОКС (FROM [станція Львів ОКС]) обрати всі записи (SELECT COUNT(*)), де маса вантажу дорівнює нулю (WHERE [Маса вантажу]=0)
Кількість порожніх критих вагонів	SELECT (COUNT* *) FROM [станція Львів ОКС] WHERE [№ ВАГОНА] 10000000=2 AND [МАСА ВАНТАЖУ]=0	З таблиці станція Львів ОКС (FROM [станція Львів ОКС]) обрати всі записи (SELECT COUNT(*)), де номер вагона починається на 2 та маса вантажу дорівнює нулю (WHERE [№ ВАГОНА] 10000000=2 AND [Маса вантажу]=0)
Сумарна маса нетто	SELECT SUM ([Маса вантажу]) AS [Сумарна маса] FROM [станція Львів ОКС]	З таблиці станція Львів ОКС (FROM [станція Львів ОКС]) обрати загальну масу вантажу (SELECT SUM ([Маса вантажу]) AS [Сумарна маса])
Дані про вагони	SELECT cdate(time( )-[час прибуття]) AS простій, * FROM [станція Львів ОКС] WHERE [№ вагона]=[№]	З таблиці станція Львів ОКС (FROM [станція Львів ОКС]) обрати різницю між поточним часом та часом прибуття в поле Простої (cdate(time( )-[час прибуття]) AS простої ), усі поля даної таблиці (*) для рядків де, номер вагона дорівнює введеному номеру вагона (WHERE [№ вагона]=[№]) *
Кількість вагонів на колії	SELECT [№> колії], COUNT(*) AS [Кількість вагонів] FROM [станція Львів ОКС] GROUP BY [№ колії]	З таблиці станція Львів ОКС (FROM [станція Львів ОКС]) обрати номер колії (SELECT [№ колії]) і кількість вагонів (COUNT(*) AS [Кількість вагонів]), згрупованих за номером колії (GROUP BY [№ колії])

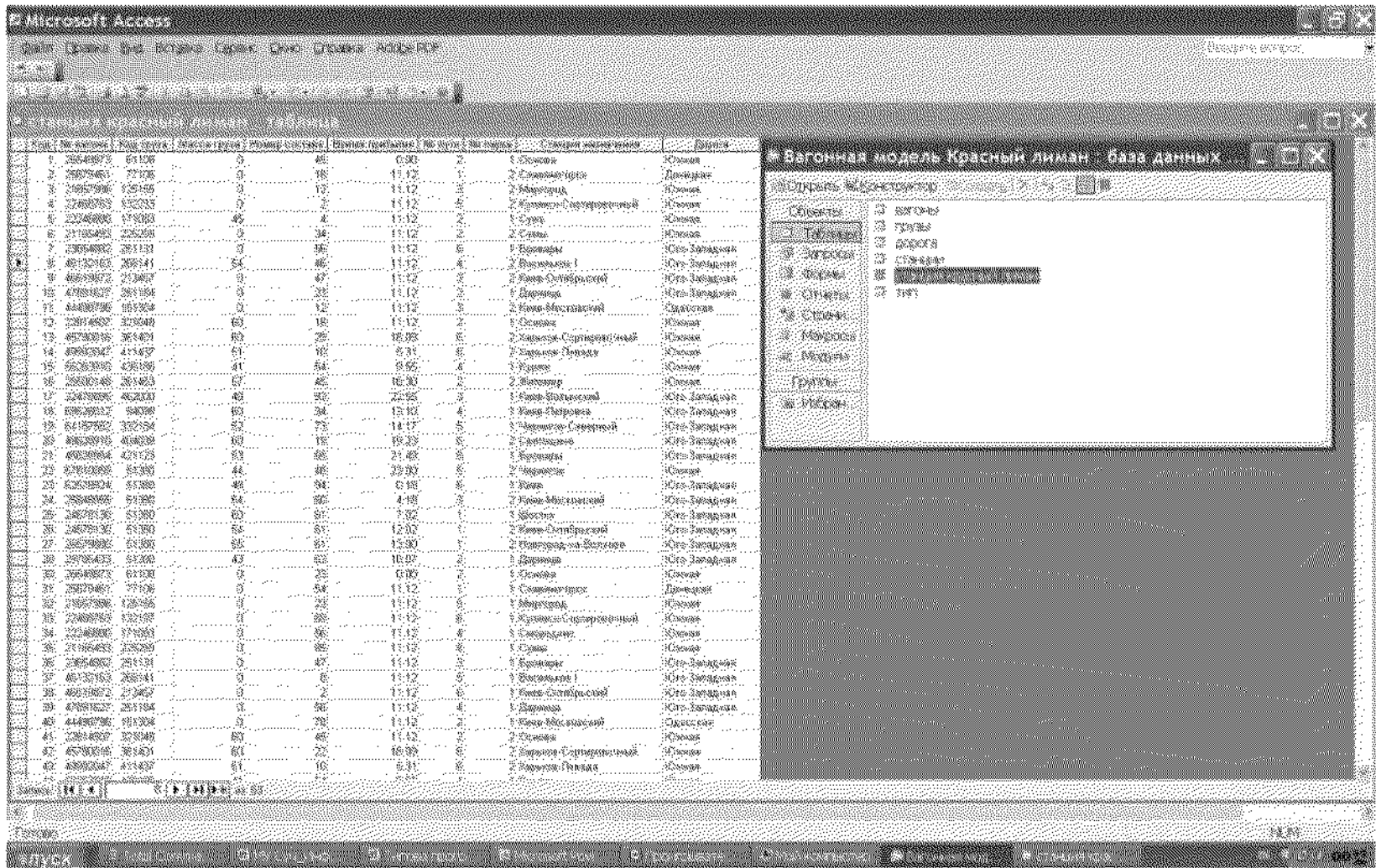


Рисунок 1.6 – Елемент бази даних «Вагонна модель станції», створений за допомогою СУБД Microsoft Access

## **2 Методи оптимізації обсягу інформації, що зберігається**

### **2.1 Поняття, пов'язані з інформаційними технологіями в галузі перевезень**

Теорія інформації (ТІ) – наука, що вивчає кількісні закономірності, пов'язані з одержанням, передачею, опрацюванням і збереженням інформації.

Для того щоб здійснити передачу інформації, вона повинна бути спочатку представлена таким чином, щоб стати зрозумілою для користувача цієї інформації. Тобто вона повинна бути подана у вигляді якихось сигналів, що сприймають органи відчуття людини (звуки, кольори, знаки, жести).

Якщо інформація має бути передана по каналу зв'язку, то ці сигнали повинні бути ще й закодовані. При цьому слід прагнути до якомога меншої кількості сигналів при збереженні обсягу інформації, що передається.

Звідси випливають два завдання ТІ:

- 1) пошук економних методів кодування інформації;
- 2) визначення кількості пристроїв, що запам'ятовують та зберігають інформацію (обсягу пам'яті, пропускної здатності каналів зв'язку).

Інформація від джерела чи від накопичувача інформації користувачу передається за допомогою повідомлень.

*Повідомлення* – це сукупність відомостей про стан деякої системи.

Повідомлення являє собою упорядкований набір сигналів, які складені з якихось окремих символів, останні в свою чергу ще закодовані за допомогою інших символів.

Повідомлення має сенс тільки в тому випадку, коли стан системи нам невідомий та ще й випадковий.

Системі  $X$  апіорі (до одержання повідомлення про її стан) властивий деякий ступінь невизначеності. Як її (невизначеність) виміряти чи обчислити?

Розглянемо три приклади.

Перший приклад. В якому стані вагон? Можливі відповіді: завантажений чи порожній. Тобто система ВАГОН має два можливі стани:

- а) вагон завантажений;
- б) вагон порожній.

Другий приклад. Якого типу вагон? Можливі відповіді: критий, піввагон, платформа, цистерна чи інші. Тобто всього система ВАГОН має 5 можливих станів.

Для кожного прикладу всі стани системи є рівноймовірними. Невизначеність системи в другому прикладі більше, тобто вона визначається кількістю можливих станів системи. Таким чином, **чим більше можливих станів має система, тим більше невизначеність системи**, але невизначеність залежить не тільки від кількості можливих станів системи.

Третій приклад. Розглянемо вагон, який може бути у двох станах:

1-й стан – вагон справний	2-й стан – вагон несправний
$P = 0,95;$	$P = 0,05.$

Така система має малу невизначеність (майже напевно вагон виявиться справним).

У прикладі із навантаженим чи порожнім вагоном, де  $P \approx 0,5$ , невизначеність системи значно більше.

**Висновок:** ступінь невизначеності системи залежить не тільки від кількості її можливих станів, але й від імовірностей цих станів.

## 2.2 Безпека та захист даних

**Класифікація (двійкових) кодів.** Коди діляться за кількістю застосованих кодових комбінацій на ненадлишкові (коли  $N = N_0$ ) і надлишкові (коли  $N < N_0$ ). Тут  $N$  – кількість застосованих кодових комбінацій.

**Ненадлишкові коди.** У ненадлишкових кодах використовується вся кількість кодових комбінацій  $N_0$ , що може бути отримана з  $n$ - двійкових розрядів ( $N_0 = 2^n$ ), тобто дорівнює кількості сполук з двох елементів по  $n$ . Тому будь-яке перекручування через перешкоди одного з розрядів (символів) викликає помилку, тому що кодове слово перетворюється у деяке інше припустиме кодове слово. У такий спосіб ненадлишкові коди не дозволяють виявляти помилки.

Ненадлишкові коди бувають **рівномірні** і **нерівномірні**.

У рівномірних кодах усі кодові комбінації мають однакову довжину і тому їх не потрібно розмежовувати (відокремлювати один від одного).

Прикладом може бути телеграфний код, за допомогою якого передають 32 символи російського алфавіту. Код має основу  $m = 2, N_0 = 32, n = \log_2 32 = 5$ .

Код із 5 двійкових символів (0,1). Усього з урахуванням цифр і знаків необхідно передавати 54 символи, тому вводять регістрові кодові комбінації, що настроюють на приймання букв або цифр.

У комп'ютерних мережах використовується восьмирозрядний двійковий код (8 біт,  $N_0 = 2^8 = 256$ ). Однак вже відчутна нестача кодових слів для кодування символів у 8-розрядних ЕОМ.

Виникає проблема національних алфавітів. 256 символів не достатньо для того, щоб кодувати усі необхідні символи повідомлень. У зв'язку з цим кодами якихось основних символів кодуються додаткові символи національних алфавітів, а при розкодуванні утворюються різночитання. Ця проблема особливо яскраво виявляється при прийманні повідомлень із Інтернету.

Запропоновано 16-бітовий код ( $N_0 = 2^{16} = 65536$ ), але поки не реалізовано.

Рівномірні коди не є оптимальними за кількістю розрядів, що припадають на один переданий символ. Як це розуміти?

Наприклад, для передачі 10 нових цифр потрібний код із чотирьох біт: 0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001. Тоді як кількість інформації, що міститься в одній десятковій цифрі

$$I = \log_2 10 = 3,32 \text{ біт}.$$

Звідси виникла ідея нерівномірних кодів, у яких символи, що часто зустрічаються, кодують більш короткими кодovими комбінаціями, а рідкісні – більш довгими (оптимальний нерівномірний код). Можна побудувати оптимальний нерівномірний код, у якому враховуються статистичні дані про частоту появи тих або інших символів у переданих повідомленнях.

Як приклад нерівномірного коду розглянемо **код Хаффмена** для кодування десяткових цифр.

0-000 2-010 4-100 6-1100(0110) 8-1110(1000)  
 1-001 3-011 5-101 7-1101(0111) 9-1111(1001).

У середньому для запису однієї десяткової цифри в цьому кодi потрібно  $(6 \cdot 3 + 4 \cdot 4) / 10 = 3,4$  біта – це майже дорівнює  $I$ .

Як правило, нерівномірні коди є **префіксними**, тобто жодна кодова комбінація не є початком іншої, що дозволяє розшифрувати повідомлення, записане без розподільвачів.

Наприклад, для коду Хаффмена виділяють два старших біти, якщо «11», то це тетрада (у кодовому слові 4 цифри), інакше – тріада (доповнюють старший 0-й біт). Тетради додатково перетворюють у двійкову систему. Рівномірні коди є теж префіксними.

**Надлишкові коди.** У надлишкових кодах загальна кількість кодових комбінацій  $N_0$  перевищує кількість кодових комбінацій  $N$ , використовуваних для запису символів повідомлення (коли  $N \leq N_0$ ). При цьому для кодування використовуються тільки такі кодові комбінації, що відрізняються не менше ніж двома розрядами. Тому



будь-яка одиночна помилка призводить до появи неприпустимої кодової комбінації, таким чином помилка буде виявлена. Можливість того, що буде перекручено два символи (подвійна помилка), значно менше.

У зв'язку з цим надлишкові коди ділять на **коди з виявленням помилок і коди з виправленням помилок**.

**Коди з виявленням помилок.** Найпростіший метод виявлення помилок – це перевірка на парність (непарність) суми символів у кодовій комбінації.

Наприклад, при передачі цифр (4 біти) можна ввести 5-й біт, значення якого визначається так, щоб кількість одиниць була непарною.

Кодове слово	Кількість одиниць
0000 1	1
0001 0	1
0010 0	1
0011 1	3
0100 0	1
0101 1	3
0110 1	3
0111 0	3
1000 0	1
1001 1	3

Загальна кількість кодових слів  $N_0 = 2^5 = 32$ , а використовується тільки  $N = 10 (N < N_0)$ .

Якщо через перешкоди при передачі змінити на протилежний будь-який символ (розряд), то кількість одиниць стане парною і помилка буде виявлена. Правда, виправити автоматично її не можна. Потрібна повторна передача.

**Коди з виправленням помилок.** Для рішення цього завдання використовують коди з виправленням помилок. Для цього множина заборонених (тих, що не використовуються) кодових слів ( $N_{затр} = N_0 - N$ ) потрібно

розбити на  $N$  підмножин, кожна з яких відображає визначену дозволена кодовою комбінацією.

Сутність виправлення полягає у тому, що при прийманні забороненої (помилкової) кодової комбінації, яка належить одній з підмножин, вона буде автоматично замінена дозволеною кодовою комбінацією, що відповідає цій підмножині.

Наприклад, 3-елементний код із кількістю дозволених кодових слів  $N = 2$  ( $A_1 = 010$ ,  $A_2 = 101$ ). Загальна кількість кодових слів  $N_0 = 2^3 = 8$ , кількість кодових комбінацій, що виправляються,  $N_{затр} = 8 - 2 = 6$ .

При одиночній помилці в  $A_1$  може бути передана одна з таких заборонених кодових комбінацій із перекручуванням одного символу:  $\{110, 000, \text{або } 011\}$ . Їм відповідає дозволена 010. В усіх цих випадках буде правильне приймання.

Відповідно при  $A_2$  й одиночній помилці (один біт перемінить значення на протилежне) одержимо іншу підмножину заборонених кодових слів  $\{001, 111, \text{або } 100\}$ , їй відповідає дозволена кодове слово 101.

Таким чином, якщо передається  $\{110, 000, 011, \text{або } 010\}$  – читається як 010, якщо  $\{001, 111, 100, 101\}$  – читається як 101. Тобто одиничні помилки не тільки виявляються, але і виправляються.

При подвійних і потрійних помилках (вони менш ймовірні) приймання повідомлення залишиться помилковим.

### **2.3 Особливості застосування інформаційних систем на залізничному транспорті України**

З погляду інформатизації залізничний транспорт є надзвичайно важким об'єктом.

До важливих особливостей залізничного транспорту відносяться такі:

- функціонування елементів транспортної системи (станції та вузли), тісний взаємозв'язок, ефективне

функціонування системи. Взаємодія між елементами можлива лише при централізованому управлінні;

- важливим елементом інформаційних систем залізничного транспорту є розвинена й високопродуктивна система передачі даних;

- значна кількість завдань управління залізничним транспортом повинна вирішуватися у режимі реального часу.

Основні принципи інформатизації: повнота і придатність інформації для користувача; точність; своєчасність; орієнтованість; гнучкість; зручність; контроль цілісності баз даних; схоронність і конфіденційність даних, захист комерційної та спеціальної інформації; адаптування системи до умов, що змінюються; прийнятність нових та існуючих версій системи; відповідний формат даних; повна інформаційна сумісність.

Стан інформатизації залізничного транспорту України на 2000 р.

1 Для задоволення інформаційних потреб в Україні функціонують: головний інформаційно-обчислювальний центр; шість інформаційно-обчислювальних центрів дорожнього рівня; 57 структурних підрозділів на залізницях; 46 передавальних пунктів; 21 АСУСС; 53 ЕОМ великої потужності; 170 ЕОМ середньої потужності; 3,5 тисячі ПЕОМ; 1635 терміналів Експерес-2.

2 Функціонування телекомунікаційних мереж: 19 тисяч кілометрів кабельної лінії зв'язку; 9 тисяч кілометрів повітряного виду зв'язку; 30 телеграфічних станцій.

3 Мережа базується на використанні такого встаткування: декадно-крокова система – 6 %; координатна система – 77 %; у базі електронних систем – 17 %.

## **3 Автоматизована система керування пасажирськими перевезеннями на залізничному транспорті України**

### **3.1 Термінальне устаткування**

Термінальне устаткування АСК ПП УЗ є універсальним устаткуванням, призначеним для виконання всіх квитково-касових операцій.

Термінали АСК ПП УЗ встановлюються у квиткових касах, бюро замовлень, ОДБ, довідкових бюро, фінансових та пасажирських службах залізниць тощо.

На термінальному устаткуванні АСК ПП УЗ забезпечена можливість: виконання квитково-касових операцій зі всіма видами проїзних документів у внутрішньодержавному та міждержавному сполученнях; отримання різних видів інформаційних довідок; отримання звітних документів про роботу касира тощо.

За функціональним призначенням термінали АСК ПП УЗ підрозділяються на: термінали квиткових кас; термінали бюро замовлень; адміністративні термінали; диспетчерські термінали; довідкові термінали; службові термінали.

За спеціалізацією термінали підрозділяються на: термінали добового продажу; термінали попереднього продажу; термінали змішаного продажу (добовий та попередній).

Терміналу може бути наданий дозвіл на оформлення безкоштовних плацкарт.

Функціональне призначення терміналів та їх спеціалізація встановлюються пасажирською службою кожної із залізниць.

Термінальне устаткування призначено для: підготовки необхідної інформації для передачі в ЕОМ системи; візуального контролю інформації, що набирається користувачем; проведення, за необхідності, коригування введеної інформації; передачі введеної інформації каналами зв'язку в обчислювальний комплекс (ОК) системи; приймання відповіді від ОК; відображення прийнятої відповіді на екрані терміналу з наступним друком

відображеної інформації на бланках проїзних і перевізних документів, квитанцій на послуги та допоміжних документів за допомогою пристрою для друку (принтера).

Термінальне устаткування квиткового касира містить такі пристрої: монітор, клавіатуру, системний блок, модем зв'язку (не для всіх терміналів), друкувальний пристрій (принтер), стабілізатор напруги.

Монітор терміналу є монітором звичайного персонального комп'ютера і призначений для відображення інформації для користувача.

Системний блок призначений для приймання замовлення, що надходить від користувача, і передачі до головної ЕОМ для оброблення, а також для приймання відповіді від головної ЕОМ та передачі повідомлення-відповіді користувачеві.

Принтер призначений для друку документів на відповідних бланках.

### 3.1.1 Пульт касира

Клавіатура пульта касира є універсальною клавіатурою персонального комп'ютера, яка призначена для введення інформації та керування роботою терміналу в процесі виконання замовлення пасажира, при набірні замовлення на отримання довідок, звітів та іншої службової інформації. Клавіатура дає можливість здійснювати набір арабських цифр та літер українського та латинського алфавітів, а також різних графічних символів. Вигляд клавіатури терміналу наведено на рисунку 3.1.

Клавіатура для зручності умовно поділяється на такі частини:

- 1) рядок функціональних клавіш (**F1**, **F2** ...);
- 2) центральну клавіатуру (літери та цифри);
- 3) клавіші керування курсором;
- 4) цифрову клавіатуру.

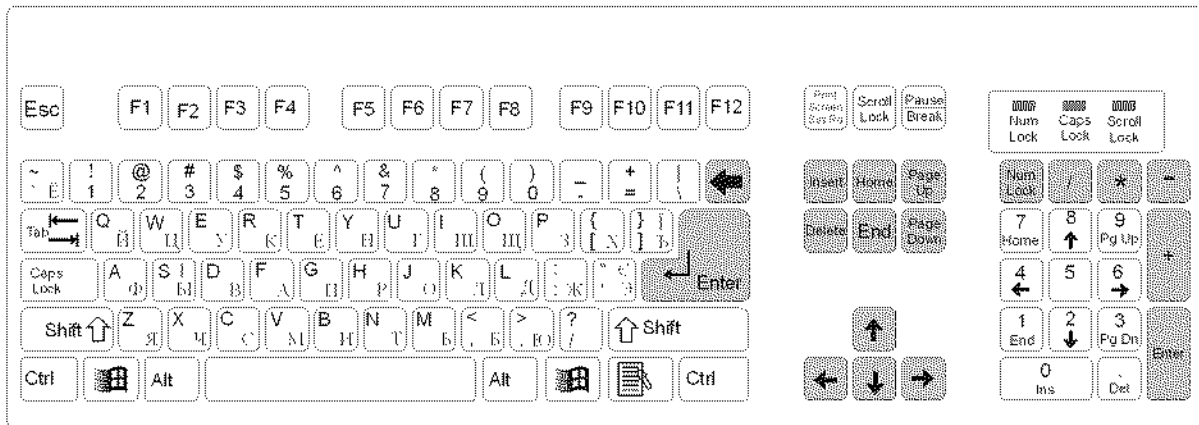


Рисунок 3.1 – Клавіатура термінала Т-УЗ

### 3.2 Нормативно-довідкова інформація

#### 3.2.1 Загальні положення щодо кодування інформації. Кодування залізниць, інформаційних статистичних центрів, залізничних адміністрацій

Для забезпечення функціонування АСК ПП УЗ вся нормативно-довідкова інформація (НДІ), що вводиться у систему, підлягає кодуванню.

У даному розділі викладені загальні принципи кодування інформації, що застосовуються при виконанні квитково-касових операцій у АСК ПП УЗ.

Кодування залізничних адміністрацій, залізниць країн СНД та Балтії, обчислювальних центрів здійснюється за єдиними принципами.

За кожною залізничною адміністрацією закріплені: двозначний цифровий код; скорочене позначення адміністрації (російське та латинське) (таблиця 3.1).

Таблиця 3.1 - Кодування залізничних адміністрацій

Назва залізничної адміністрації	Код	Скорочене позначення	
Російські залізниці	20	РЖД	RZD
Білоруська залізниця	21	БЧ	BC
Укрзалізниця	22	УЗ	UZ
Молдавська залізниця	23	ЧФМ	CMF
Литовська залізниця	24	ЛГ	LG
Латвійська залізниця	25	ЛДЗ	LZD
Державне підприємство „Естонська залізниця”	26	ЭВР	EVR
Казахські залізниці	27	КЗХ	KZH
Грузинська залізниця	28	ГР	GRZ
Азербайджанська залізниця	57	АЗ	AZ
Вірменська залізниця	58	АРМ	ARM
Киргизька залізниця	59	КРГ	KRG
Таджицька залізниця	66	ТДЖ	TZD
Туркменська залізниця	67	ТРК	TRK

Кодування залізниць (Таблиця 3.2) здійснюється за допомогою скороченої назви. Кодування ІСЦ виконується одним символом.

Таблиця 3.2 – Кодування залізниць

Назва залізниці*	Скорочене позначення залізниці	Літерний код ОЦ
АЗЕРБАЙДЖАНСКАЯ	АЗ	-
АРМЯНСКАЯ	АРМ	-
БЕЛОРУССКАЯ	БЧ	Б
ВОСТОЧНО - СИБИРСКАЯ	В-СИБ	И
ГОРЬКОВСКАЯ	ГОР	Г
ГРУЗИНСКАЯ	ГРЗ	-
ДОНЕЦЬКА	ДОН	Д
ДАЛЬНЕВОСТОЧНАЯ	Д-В	И
ЗАБАЙКАЛЬСКАЯ	ЗБК	И
ЗАПАДНО-СИБИРСКАЯ	З-СИБ	Н
КАЛИНИНГРАДСКАЯ	КЛГ	М
КУЙБЫШЕВСКАЯ	КБШ	Й
КЫРГИЗСКАЯ	КРГ	Ц
ЛАТВИЙСКАЯ	ЛДЗ	Р
ЛИТОВСКАЯ	ЛГ	Р
ЛЬВІВСЬКА	ЛЬВ	К
МОЛДАВСКАЯ	ЧФМ	Ь
МОСКОВСКАЯ	МСК	М
ОДЕСЬКА	ОДС	Ш
ОКТЯБРЬСКАЯ	ОКТ	О
ПРИВОЛЖСКАЯ	ПРИВ	Й
ПРИДНІПРОВСЬКА	ПРИДН	П
СВЕРДЛОВСКАЯ	СВРД	Е
СЕВЕРНАЯ	СЕВ	Я
СЕВЕРО - КАВКАЗСКАЯ	С-КВ	С
УЗБЕКСКАЯ	УЗБК	Ф
ТУРКМЕНСКАЯ	ТРК	-
КАЗАХСКАЯ	КЗХ	Ц
ЭСТОНСКАЯ	ЭВР	Р
ЮГО-ВОСТОЧНАЯ	Ю-В	С
ПІВДЕННО-ЗАХІДНА	Ю-З	К
ПІВДЕННА	ЮЖ	Ю
ЯКУТСКАЯ	ЖДЯ	И
САХАЛИНСКАЯ	САХ	И
ТАДЖИКСКАЯ	ТЖД	-

\* Назви залізниць України зазначені українською мовою. Назви інших залізниць, що наведені в таблиці, зазначені російською мовою



### 3.2.2 Кодування поїздів та ниток

Для АСК ПП УЗ прийнята п'ятизначна нумерація поїздів – три арабські цифри та дві літери кирилиці (рисунок 3.2).

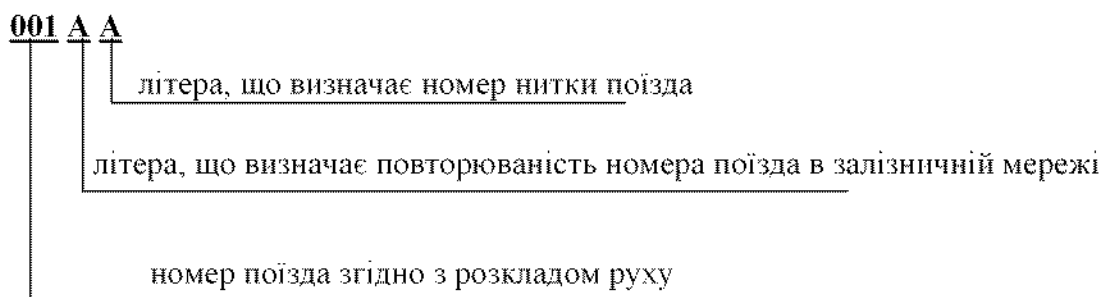


Рисунок 3.2 – Нумерація поїздів, прийнята для АСК ПП УЗ

Необхідність введення літери повторюваності обумовлена тим, що на мережі залізниць в обігу може бути декілька поїздів з однаковим цифровим номером. Літера повторюваності однозначно визначає поїзд, тому при введенні замовлення вказується обов'язково.

П'ятизначна нумерація поїздів дозволяє кодувати всі можливі маршрути прямування поїздів з урахуванням причіпних вагонів та ниток поїзда.

Номер нитки визначає сукупність місць в одному або декількох вагонах поїзда, що прямують одним безпересадковим маршрутом. При введенні замовлення на поїздку номер нитки не вказується. На заповненому бланку проїзного документа зазначається номер нитки.

### 3.2.3 Кодування вагонів

В АСК ПП УЗ номер вагона позначається арабськими цифрами (від 1 до 3). Тризначна нумерація вагонів використовується тільки у поїздах міжнародного сполучення.

Коди касир набирає при введенні замовлення на поїзду. Аналогічні позначення типу вагона також вказуються на проїзному документі.

Кодування типів вагонів (крім швидкісних поїздів, електропоїздів та дизель-поїздів) здійснюється за допомогою літер кирилиці.

Тип вагона	Позначення
Люкс	Л
М'який	М
Купейний	К
Плацкартний	П
Загальний	О
Міжобласний	С

Для швидкісних поїздів, електропоїздів та дизель-поїздів зазначення типу вагона в замовленні касира здійснюється за наведеною схемою.

Тип вагона	Позначення
Сидячі вагони 1-го класу	К
Сидячі вагони 2-го класу	П
Сидячі вагони 3-го класу	О

У цих випадках на проїзному документі у першому рядку вказується номер вагона завжди з літерою С (сидячий); у другому рядку (після кодів станцій відправлення та призначення пасажирів) вказується відповідний клас вагона.

### 3.2.4 Кодування станцій

Для позначення станцій у системі використовується єдина міжнародна цифрова нумерація. Кодування станцій здійснюється за допомогою арабських цифр (рисунок 3.3).

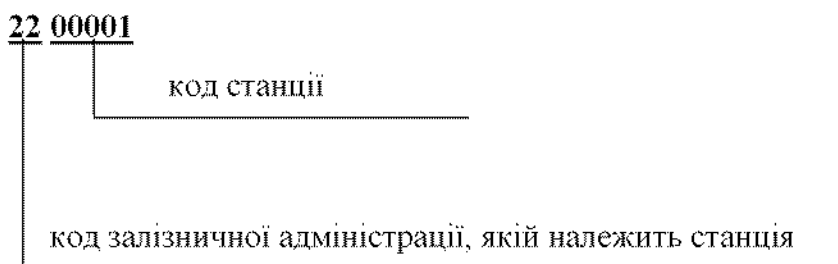


Рисунок 3.3 – Кодування станцій

Третій та четвертий розряди у кодї станції призначені для кодування залізниць у межах однієї залізничної адміністрації, якщо цих залізниць декілька. Якщо в межах залізничної адміністрації існує одна залізниця, то третій та четвертий розряди використовуються на розсуд служб конкретної залізничної адміністрації.

Розряди з 5 по 7 – номер станції.

Алфавітні назви станцій російською та українською мовами (не більше 12 символів) повинні відповідати класифікатору станцій.

### 3.2.5 Регулювання продажу місць у поїздах

Для здійснення регулювання продажу місць у поїздах в АСК ПП УЗ існують такі способи:

- 1) встановлення режимів продажу місць у нитках поїзда: “зима”, “весна-осінь”, “літо”;
- 2) виділення норми місць у трафаретних нитках поїзда;
- 3) встановлення регулювання продажу “від” та “до” станцій маршруту поїзда;
- 4) присвоєння станціям маршруту поїзда ознак “станція попереднього продажу”;
- 5) виділення місць у поїздах для організації продажу у касах проміжних станцій та касах залізниць СНД, Балтії та Європи, організації зворотнього виїзду, організації групових перевезень дітей, організації оперативного регулювання довжини поїздів тощо.

Встановлення режиму “зима” забезпечує можливість здійснення попереднього та добового продажу місць від

будь-якої станції маршруту поїзда всіма касами незалежно від їхнього місцезнаходження.

Встановлення режиму “весна-осінь” забезпечує можливість здійснення попереднього продажу тільки від початкової станції маршруту всіма касами незалежно від їхнього місцезнаходження; поточний продаж від будь-якої станції маршруту поїзда здійснюється тільки після відправлення поїзда з початкової станції маршруту.

Встановлення режиму “літо” забезпечує можливість здійснення попереднього продажу тільки від початкової станції маршруту всіма касами незалежно від їх місцезнаходження; поточний продаж від будь-якої станції маршруту поїзда здійснюється тільки після відправлення поїзда з попередньої станції маршруту.

Визначення режимів продажу місць у нитках поїздів здійснює пасажирська служба залізниці відправлення поїзда.

Виділення норми місць у трафаретних вагонах поїзда забезпечує можливість здійснення попереднього та добового продажу місць у транзитних поїздах для початкової станції трафаретної нитки.

Виділення норми місць у трафаретних вагонах поїзда здійснюється за узгодженням пасажирськими службами залізниць.

Встановлення регулювання продажу “від” та “до” (“від – до”) станцій маршруту поїзда забезпечує можливість регулювання пасажиропотоків до (від) певних станцій з урахуванням можливості альтернативного проїзду до (від) цих станцій іншими поїздами.

Визначення необхідності встановлення регулювання продажу “від” та “до” (“від - до”) станцій маршруту поїзда здійснює пасажирська служба залізниці відправлення поїзда.

Присвоєння станціям маршруту поїзда ознак “Станція попереднього продажу” забезпечує можливість організації попереднього та поточного продажу з проміжних станцій у разі встановлення режимів “весна-осінь” та “літо”.

Визначення станцій попереднього продажу здійснює пасажирська служба залізниці.

### **3.3 Квитково-касові операції**

#### **3.3.1 Загальні положення**

Квитково-касові операції містять у собі:

- 1) оформлення проїзних документів у квиткових касах, касах бюро замовлень та інше;
- 2) бронювання місць для індивідуальних пасажирів;
- 3) резервування місць для організованих груп пасажирів;
- 4) погашення зіпсованих проїзних документів у квиткових касах, касах бюро замовлень та інше;
- 5) повернення невикористаних проїзних документів;
- 6) переоформлення проїзних документів;
- 7) отримання звітів касира в касах, касах бюро замовлень та інше;
- 8) отримання інформаційних довідок;
- 9) отримання допоміжних документів та довідок у бюро замовлень;
- 10) отримання довідок про роботу термінала;
- 11) оформлення, погашення та повернення квитанцій на послуги;
- 12) оформлення, погашення та повернення перевізних документів для перевезення багажу “на руках”;
- 13) резервування місць у міжнародному сполученні.

Касир повинен враховувати вимоги пасажира до поїздки.

#### **4 Автоматизована система керування вантажними перевезеннями на залізничному транспорті України**

Останнім часом в усьому світі обчислювальні і програмні засоби бурхливо розвиваються, сприяючи розвитку промисловості. Нові економічні умови господарювання у залізничній галузі вимагають рішення нових комплексних завдань, які практично неможливо вирішити в рамках існуючих систем. До таких завдань можна віднести, наприклад:

- ✓ контроль за просуванням окремої відправки;
- ✓ керування термінами доставки вантажів;
- ✓ оперативний контроль за станом взаєморозрахунків клієнтів;
- ✓ керування технолого-економічною моделлю перевізного процесу;
- ✓ аналіз та керування вантажопотоками та інш.

Для вирішення цих і багатьох інших завдань на мережі УЗ розробляється та впроваджується автоматизована система керування вантажними перевезеннями на залізничному транспорті України (АСК ВП УЗ) з використанням новітніх і перспективних розробок. Здійснюється принциповий перехід від окремих АС до єдиної інтегрованої бази даних галузі.

Джерелом зародження інформації для вирішення зазначених завдань є товарні контори залізниць, де проходить “народження перевізних документів”. Комплект перевізних документів (КПД) є суттєво первинним та повною мірою інформативно достатнім, з якого розпочинається процес перевезення. Саме тому розвиток АСК ВП УЗ починається з автоматизації роботи товарних касирів – АРМ ТВК.

АРМ ТВК є підсистемою АСК ВП УЗ, яка виконує завдання лінійного рівня. Воно забезпечує зручний інтерфейс користувача для формування електронного документа, передачі його до бази даних для подальшого оброблення та аналізу.

Максимальна достовірність інформації, яка вводиться у єдину базу перевізних документів (на сервер), забезпечується максимальною кількістю логічних контролів за кожним видом і типом інформації, що вводиться.

АРМ ТВК дозволяє виконувати такі операції:

- ✓ роздрукувати комплект перевізних документів (можливий друк тільки нарахованих платежів);

- ✓ виконати сервісне обслуговування клієнтури з надання попереднього таксування;

- ✓ підготуватися до введення великої кількості одноманітних КПД, склавши на них шаблони;

- ✓ отримати інформацію про динамічне сальдо розрахункових рахунків клієнтури;

- ✓ спростити роботу товарного касира після прибуття за рахунок передачі електронного документа в пункт призначення;

- ✓ скоротити термін надання документів з товарної контори на ТехПД (декілька секунд);

- ✓ виключити повторне введення документів на ТехПД, після його надання;

- ✓ підвищити пропускну здатність товарної контори;

- ✓ покращити якість обслуговування клієнтів і т.д.

АРМ ТВК працює у взаємодії з серверною частиною системи, яка забезпечує все оброблення перевізних документів, – сервером вантажної роботи АСК ВП УЗ (СВР).

#### **4.1 Основні відомості**

Автоматизоване робоче місце інженера з ТРА (АРМ ТРА) – це комплекс технічних та програмних засобів, що має забезпечити за рахунок використання сучасних інформаційних технологій швидке та якісне створення, ведення і формування технічно-розпорядчих актів (ТРА) роздільних пунктів.

Автоматизоване робоче місце з розрахунку норм закріплення рухомого складу гальмовими башмаками на станційних коліях (АРМ РНЗ) – це комплекс технічних та програмних засобів, що має забезпечити проведення

швидкого та якісного розрахунку норм закріплення і заповнення на базі проведених розрахунків пунктів 3.10 (27) ТРА.

Дана технологія містить опис організаційної структури завдань АРМ ТРА та АРМ РНЗ, а також порядок забезпечення їх взаємодії.

У документі визначено розподіл обов'язків і відповідальності персоналу, передбачено порядок надання інформації користувачам структурних підрозділів УЗ.

## **4.2 Технологія функціонування**

АРМ ТРА та АРМ РНЗ функціонують на двох рівнях: серверний та клієнтський.

Введення даних проводиться на ПЕОМ користувачів клієнтського рівня, вся введена інформація передається на сервер по каналах зв'язку, оброблюється на сервері та надається працівниками залізниць (користувачам серверного рівня) у вигляді зручних відеограм.

### **4.2.1 Функції серверної частини**

Для централізованого обміну інформацією використовується інтранет-сервер ТРА УЗ. На етапі дослідної експлуатації програмного комплексу інтранет-сервер знаходиться в ІСЦ Донецької залізниці. Адреса інтранет-сервера: <http://10.7.224.23/armtra>

Сервер має підключення до СУБД Oracle. Всі операції із адміністрування виконують спеціалісти ІСЦ Донецької залізниці.

Інтранет-сервер виконує такі функції:

- централізоване зберігання файлів даних ТРА та схем графічних додатків в обсязі УЗ в СУБД Oracle;
- забезпечення безперервного обміну інформацією (операції імпорту та експорту) між сервером та робочими місцями інженерів з ТРА із веденням відповідних журналів та статистики;



- надання користувачам керівного складу необхідної статистики щодо обсягів наповнення інформаційної бази ТРА у вигляді зручних відеограм;
- зберігання та надання користувачам АРМ ТРА та АРМ РНЗ інсталяційних пакетів програмного комплексу, останніх версій програмних поновлень, НДІ-класифікаторів;
- надання працівникам УЗ (диспетчерам, керівному складу та ін.) зручного інтерфейсу для отримання актуальної інформації з ТРА до графічних додатків;
- розмежування доступу користувачів до інформації залежно від потреб.

#### **4.2.2 Функції клієнтської частини**

Клієнтська частина АРМ ТРА та АРМ РНЗ являє собою інсталяційний пакет, який містить такі програми та документи:

1 Програмні модулі:

а) головний модуль АРМ ТРА (у тому числі субмодулі роботи з сервером імпорту, експорту та поновлення програмних файлів);

б) модулі редакторів оснащення роздільних пунктів;

в) модуль виконання розрахунків норм закріплення.

2 Набір НДІ-класифікаторів (кількість – близько 100), які використовуються у процесі ведення ТРА та при проведенні розрахунків норм.

3 Набір шаблонів для формування вихідних документів – пунктів ТРА та результатів розрахунків норм закріплення.

4 Довідникові електронні документи «АРМ ТРА. Керівництво користувача», «АРМ ТРА. Керівництво програміста», «Технологія функціонування АРМ ТРА та АРМ РНЗ в межах УЗ».

### 4.2.3 Методика взаємодії АРМ ТРА та АРМ РНЗ

Технологічні процеси автоматизованого ведення ТРА станцій та виконання розрахунків норм закріплення за допомогою АРМ РНЗ тісно пов'язані між собою.

Взаємодія двох програмних комплексів полягає у послідовному виконанні таких операцій:

1 Введення в АРМ ТРА інформації щодо станційних колій, колій структурних підрозділів залізниць та під'їзних колій.

2 Доповнення цієї інформації в АРМ РНЗ даними масштабних (схематичних) планів або масштабних схем щодо поздовжніх профілів цих колій.

3 Проведення в АРМ РНЗ всіх необхідних розрахунків.

4 Експортування результатів проведених розрахунків із АРМ РНЗ по станційних коліях у пункти 3.10 форми ДУ-41 та 27 форми ДУ-41а ТРА, по під'їзних коліях у відомості під'їзних колій форми ДУ-41б.

Забезпечити взаємодію та коректний обмін даними між двома програмними комплексами дозволяють ряд операцій з інтеграції АРМ РНЗ в АРМ ТРА, проведених розробниками спільного програмного комплексу:

- запуск АРМ РНЗ виконується безпосередньо із програмного середовища АРМ ТРА в контексті обраного роздільного пункту;

- АРМ РНЗ та АРМ ТРА спільно використовують одні й ті ж самі файли даних роздільних пунктів, класифікаторів і т.д.;

- дані файлів роздільних пунктів попередньої версії АРМ РНЗ придатні для використання у новому спільному комплексі;

- вихідні дані (результати розрахунків) АРМ РНЗ надаються у форматі, придатному для коректного перенесення в АРМ ТРА;

- програмний комплекс орієнтований на виконання завдань АРМ ТРА та АРМ РНЗ у межах конкретного роздільного пункту одним користувачем.

## Контрольні питання з дисципліни

### Основні питання

- 1 Основні принципи побудови АСОУП.
- 2 Основні принципи побудови ДИСПАРК.
- 3 Основні принципи побудови ДИСЛОК.
- 4 Основні принципи побудови АСУ СС.
- 5 Основні дані, що містяться у ТГНЛ, та призначення ТГНЛ.
- 6 Об'єкти кодування на залізничному транспорті.
- 7 Код станції в АСУ ЗТ.
- 8 Контрольний знак коду станції.
- 9 Мережний район.
- 10 Код залізниці.
- 11 Номер вагона.
- 12 Контрольний знак номера вагона.
- 13 Код вантажу.
- 14 Код вантажовідправника і вантажоодержувача.
- 15 Номер поїзда .
- 16 Інформаційне повідомлення в АСОУП.
- 17 Службове повідомлення в АСОУП.
- 18 Коригувальне повідомлення в АСОУП.
- 19 Повідомлення-запит в АСОУП.
- 20 Макет повідомлення в АСОУП.
- 21 Скільки службових блоків включають повідомлення в АСОУП?
- 22 Скільки інформаційних блоків включають повідомлення в АСОУП?

### Додаткові питання

- 1 Комп'ютер по відношенню до людини – це ...
- 2 Основні складові комп'ютера.
- 3 Основні складові системного блока комп'ютера.
- 4 Що є витратним матеріалом у матричному принтері?
- 5 Що є витратним матеріалом у струминному принтері?
- 6 Що є витратним матеріалом у лазерному принтері?

7 Передавати зображення до комп'ютера – це основне призначення...

8 «Розпізнати текст» означає ...

9 По яких двох середовищах може передаватися сигнал від одного комп'ютера до іншого?

## **Список літератури**

### **Основна література**

1 Закон України “Про залізничний транспорт” // Магістраль. – 1996. – №56 (або “Основні нормативні акти у галузі залізничного транспорту України”. – Харків: ХарДАЗТ, 2000).

2 Концепция и программа реструктуризации на ж.-д. транспорте Украины // Магістраль. – 1997. – №5.

3 Правила перевезень вантажів: Збірники ППТЗТ України. № 1 – 9 та інші. – К., 1997. – 2002.

4 Смехов А. А. Маркетинговые модели транспортного рынка. – М.: Транспорт, 1998.

5 Глушаков С.В., Ломотько Д.В. Microsoft SQL Server: Учебный курс. – Харьков: Фолио, 2000. – 498 с.

6 Глушаков С.В., Ломотько Д.В., Сурядный А.С. Работа в сети Internet: Учебный курс. – Харьков: Фолио, 2000. – 330 с.

7 Рынок и логистика: Сборник. – М.: Экономика, 1992.

8 Транспортная логистика: Учебник / Под ред. проф. Л.Б. Миротина. – М.: Экзамен, 2003.

### **Додаткова література**

9 Логистика: Учеб. пособие / Под ред. Б.А. Аникина. – М.: ИНФРА-М, 1997. – 327 с.

10 Гаджинский А.М. Основы логистики. – М.: ИВЦ “Маркетинг”, 1998. – 228 с.

11 Первый семинар по проблемам логистики // Подъемно-транспортная техника и склады. – 1990. – № 1. – С. 35.

12 Принципы логистики и тароупаковочное хозяйство // Подъемно-транспортная техника и склады. – 1989. – № 2. – С. 73.

13 Ходош М.С. Грузовые автомобильные перевозки. - М.: Транспорт, 1986.

14 Плоткин Б.К. Основы логистики. - Львов: Политехника, 1991. – 112 с.

15 Новиков О.А., Семенко А.И. Производственно-коммерческая логистика. Ч. 1 и 2. – СПб.: Изд-во С.-П. унив-та экономики и финансов, 1993. – 208 с.

16 Транспортная тара: Справочник / А.И. Телегин, Ю.А. Балберов, Н.И. Денисов и др. – М.: Транспорт, 1996. – 216 с.

17 Гордон М.П., Тшикин Е.М., Усков Н.С. Как осуществить экономическую доставку товара отечественному и зарубежному покупателю: Справочное пособие для предпринимателя. – М.: Транспорт, 1993. – 64 с.

18 Гордон М.П. Функции и развитие логистики в сфере товарообращения // РИСК. – 1993. – №1. – С. 125.

19 Логистика: Учеб. пособие / О.Г. Туровец, В.Н. Родионова. – Воронеж: Воронеж. гос. техн. ун-т, 1994. – 90 с.

