

## 3.1 Вставка, видалення і оновлення даних / Insert, delete, and update data

### **REPLACE**

додавання або заміщення рядків таблиці / inserts or  
updated table records

**REPLACE INTO** <table\_name>

[(<columns\_list>)]

**VALUES**

(<values\_list\_1>),

(<values\_list\_2>),

...

(<values\_list\_N>);

## 3.1 Вставка, видалення і оновлення даних / Insert, delete, and update data

### **REPLACE == INSERT**

значення первинного ключа або унікального індексу  
не збігається ні з одним з вже існуючих значень  
the value of the primary key or unique index does not  
match any of the existing values

в іншому випадку, перед додаванням нового рядка  
колишній рядок видаляється  
otherwise, the previous row is deleted before adding a  
new row

### 3.1 Вставка, видалення і оновлення даних / Insert, delete, and update data

REPLACE INTO supplied  
VALUES  
(4, 'Printer', 41, 332.50);

- не можна задавати нові значення, що обчислюються з використанням колишніх значень / you can not set new values using previous values
- інакше буде підставлене значення за замовчуванням / otherwise, a default value will be set

### 3.1 Вставка, видалення і оновлення даних / Insert, delete, and update data

#### **DELETE**

Видалення рядків таблиці / Remove table records

**DELETE FROM** <table>

[**WHERE** <condition>]

[**ORDER BY** <column\_name> [**ASC** | **DESC**]]

[**LIMIT** <rows\_number>];

DELETE FROM supplied WHERE contract\_number = 5;

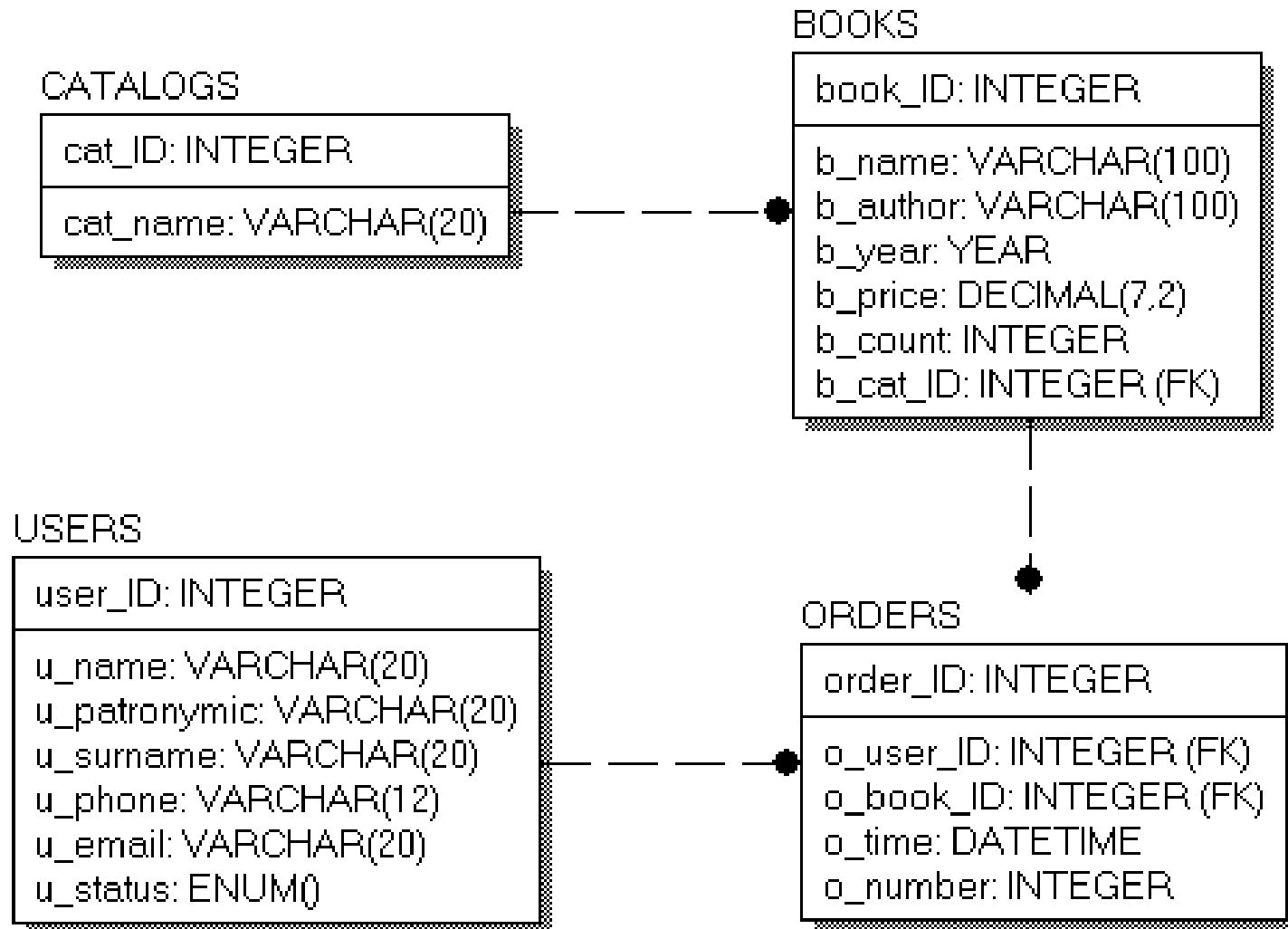
## 3.1 Вставка, видалення і оновлення даних / Insert, delete, and update data

Операції вставки, видалення і зміни рядків  
таблиці

Operations used to insert, delete, and modify  
table rows

- INSERT
- UPDATE
- REPLACE
- DELETE

### 3.1 Вставка, видалення і оновлення даних / Insert, delete, and update data



## 3.1 Вставка, видалення і оновлення даних / Insert, delete, and update data

```
DROP DATABASE IF EXISTS book;

CREATE DATABASE book
CHARACTER SET utf8
COLLATE utf8_general_ci;

USE book;

CREATE TABLE catalogs (
    cat_ID int(6) NOT NULL
    AUTO_INCREMENT,
    cat_name varchar(20) NOT NULL,
    PRIMARY KEY (cat_ID)
) ENGINE=InnoDB;

CREATE TABLE books (
    book_ID int(6) NOT NULL
    AUTO_INCREMENT,
    b_name varchar(100) NOT NULL,
    b_author varchar(100) NOT NULL,
    b_year year NOT NULL,
    b_price decimal(7,2) NULL default
    '0.00',
    b_count int(6) NULL default '0',
    b_cat_ID int(6) NOT NULL default '0',
    PRIMARY KEY (book_ID),
    FOREIGN KEY (b_cat_ID) REFERENCES
    catalogs (cat_ID) ON DELETE
    CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB;
```


## 3.1 Вставка, видалення і оновлення даних / Insert, delete, and update data


```
CREATE TABLE users (  
    user_ID int(6) NOT NULL  
    AUTO_INCREMENT,  
    u_name varchar(20) NOT NULL,  
    u_patronymic varchar(20) NOT NULL,  
    u_surname varchar(20) NOT NULL,  
    u_phone varchar(12) NULL,  
    u_email varchar(20) NULL,  
    u_status ENUM  
    ('active','passive','lock','gold')  
    default 'passive',  
    PRIMARY KEY (user_ID)  
) ENGINE=InnoDB;
```

```
CREATE TABLE orders (  
    order_ID int(6) NOT NULL  
    AUTO_INCREMENT,  
    o_user_ID int NOT NULL,  
    o_book_ID int NOT NULL,  
    o_time datetime NOT NULL default  
    '0000-00-00 00:00:00',  
    o_number int(6) NOT NULL default '0',  
    PRIMARY KEY (order_ID),  
    FOREIGN KEY (o_book_ID)  
    REFERENCES books(book_ID) ON  
    DELETE CASCADE ON UPDATE CASCADE,  
    FOREIGN KEY (o_user_ID)  
    REFERENCES users (user_ID) ON  
    DELETE CASCADE ON UPDATE CASCADE  
) ENGINE=InnoDB;
```



### 3.1 Вставка, видалення і оновлення даних / Insert, delete, and update data

catalogs (2×5)		catalogs (2×2)	
 cat_ID	cat_name		
1	Программирование		
2	Интернет		
3	Базы данных		
4	Сети		
5	Мультимедиа		

catalogs (2×5)		catalogs (2×2)	
 cat_ID	cat_name		
1	Программирование		
2	Интернет		

```
SELECT * FROM catalogs;
```

```
DELETE FROM catalogs WHERE cat_ID > 2;
```

```
SELECT * FROM catalogs;
```

## 3.1 Вставка, видалення і оновлення даних / Insert, delete, and update data

catalogs (2×5)		catalogs (2×0)	
cat_ID	cat_name		
1	Программирование		
2	Интернет		
3	Базы данных		
4	Сети		
5	Мультимедиа		




```
SELECT * FROM catalogs;
```

```
DELETE FROM catalogs;
```

```
SELECT * FROM catalogs;
```

catalogs (2×5)		catalogs (2×0)	
cat_ID	cat_name		




## 3.1 Вставка, видалення і оновлення даних / Insert, delete, and update data

orders (5×5)		orders (5×2)			
 order_ID	 o_user_ID	 o_book_ID	o_time	o_number	
1	3	8	2009-01-04 10:39:38	1	
2	6	10	2009-02-10 09:40:29	2	
3	1	20	2009-02-18 13:41:05	4	
4	4	20	2009-03-10 18:20:00	1	
5	3	20	2009-03-17 19:15:36	1	




```
SELECT * FROM orders;
```

```
DELETE FROM orders LIMIT 3;
```

```
SELECT * FROM orders;
```

orders (5×5)		orders (5×2)			
 order_ID	 o_user_ID	 o_book_ID	o_time	o_number	
4	4	20	2009-03-10 18:20:00	1	
5	3	20	2009-03-17 19:15:36	1	



## 3.1 Вставка, видалення і оновлення даних / Insert, delete, and update data

orders (5×5)		orders (5×2)			
 order_ID	 o_user_ID	 o_book_ID	o_time	o_number	
1	3	8	2009-01-04 10:39:38	1	
2	6	10	2009-02-10 09:40:29	2	
3	1	20	2009-02-18 13:41:05	4	
4	4	20	2009-03-10 18:20:00	1	
5	3	20	2009-03-17 19:15:36	1	

```
SELECT * FROM orders;
```

```
TRUNCATE TABLE orders;
```

```
SELECT * FROM orders;
```

orders (5×5)		orders (5×0)	
 order_ID	 o_user_ID	 o_book_ID	<div>o_time</div> <div>o_number</div>

### 3.1 Вставка, видалення і оновлення даних / Insert, delete, and update data

catalogs (2×5)	
cat_ID	cat_name
1	Программирование
2	Интернет
3	Базы данных
4	Сети
5	Мультимедиа

catalogs (2×5)	
cat_ID	cat_name
1	Программирование
2	Интернет
3	Базы данных
4	Компьютерные сети
5	Мультимедиа

```
SELECT * FROM catalogs;
```

```
UPDATE catalogs SET cat_name = 'Комп'ютерні мережі'  
WHERE cat_name = 'Мережі';
```

```
SELECT * FROM catalogs;
```

## 3.1 Вставка, видалення і оновлення даних / Insert, delete, and update data

books (7×30)		books (7×30)				
book_ID	b_name	b_author	b_year	b_price	b_count	b_cat_ID
1	JavaScript в кармане	Рева О.Н.	2008	42.00	10	1
2	Visual FoxPro 9.0	Клепинин В.Б.	2007	660.00	2	1
3	C++ Как он есть	Тимофеев В.В.	2009	218.00	4	1
4	Создание приложений с помощью C#	Фаронов В.В.	2008	169.00	1	1
5	Delphi. Народные советы	Шкрыль А.А.	2007	243.00	6	1
6	Delphi. Полное руководство	Сухарев М.	2008	500.00	6	1
7	Профессиональное программирование на PHP	Шлоссейгл Дж.	2006	309.00	5	1
8	Совершенный код	Макконнелл С.	2007	771.00	1	1
9	Практика программирования	Керниган Б.	2004	214.00	12	1

```
SELECT * FROM books;
```

```
UPDATE books SET b_price = b_price * 0.95;
```

```
SELECT * FROM books;
```

books (7×30)		books (7×30)				
book_ID	b_name	b_author	b_year	b_price	b_count	b_cat_ID
1	JavaScript в кармане	Рева О.Н.	2008	39.90	10	1
2	Visual FoxPro 9.0	Клепинин В.Б.	2007	627.00	2	1
3	C++ Как он есть	Тимофеев В.В.	2009	207.10	4	1
4	Создание приложений с помощью C#	Фаронов В.В.	2008	160.55	1	1
5	Delphi. Народные советы	Шкрыль А.А.	2007	230.85	6	1
6	Delphi. Полное руководство	Сухарев М.	2008	475.00	6	1
7	Профессиональное программирование на PHP	Шлоссейгл Дж.	2006	293.55	5	1
8	Совершенный код	Макконнелл С.	2007	732.45	1	1
9	Практика программирования	Керниган Б.	2004	203.30	12	1

## 3.1 Вставка, видалення і оновлення даних / Insert, delete, and update data

books (7×30)		books (7×30)				
book_ID	b_name	b_author	b_year	b_price	b_count	b_cat_ID
1	JavaScript в кармане	Рева О.Н.	2008	42.00	10	1
2	Visual FoxPro 9.0	Клепинин В.Б.	2007	660.00	2	1
3	C++ Как он есть	Тимофеев В.В.	2009	218.00	4	1
4	Создание приложений с помощью C#	Фаронов В.В.	2008	169.00	1	1
5	Delphi. Народные советы	Шкрыль А.А.	2007	243.00	6	1
6	Delphi. Полное руководство	Сухарев М.	2008	500.00	6	1
7	Профессиональное программирование на PHP	Шлоссейгл Дж.	2006	309.00	5	1
8	Совершенный код	Макконнелл С.	2007	771.00	1	1
9	Практика программирования	Керниган Б.	2004	214.00	12	1

```
SELECT * FROM books;
```

```
UPDATE books SET b_price = b_price * 0.95, b_count = b_count - 1;
```

```
SELECT * FROM books;
```

books (7×30)		books (7×30)				
book_ID	b_name	b_author	b_year	b_price	b_count	b_cat_ID
1	JavaScript в кармане	Рева О.Н.	2008	39.90	9	1
2	Visual FoxPro 9.0	Клепинин В.Б.	2007	627.00	1	1
3	C++ Как он есть	Тимофеев В.В.	2009	207.10	3	1
4	Создание приложений с помощью C#	Фаронов В.В.	2008	160.55	0	1
5	Delphi. Народные советы	Шкрыль А.А.	2007	230.85	5	1
6	Delphi. Полное руководство	Сухарев М.	2008	475.00	5	1
7	Профессиональное программирование на PHP	Шлосснейгл Дж.	2006	293.55	4	1
8	Совершенный код	Макконнелл С.	2007	732.45	0	1
9	Практика программирования	Керниган Б.	2004	203.30	11	1

## 3.2 Створення запитів на вибірку / Creating select queries

### **SELECT**

команда для отримання даних з таблиць бази даних

the command used to retrieve data from database tables

- вивести всі дані таблиці / display all table data

**SELECT \* FROM** <table\_name>;




## 3.2 Створення запитів на вибірку / Creating select queries

- можна вказати список стовпців таблиці / it is possible to specify a list of table columns

**SELECT** <column\_name\_1>, ..., <column\_name\_N> **FROM** <table\_name>;

SELECT \* FROM supplier;

SELECT supplier\_id, supplier\_address, supplier\_phone FROM supplier;

supplier (3×5)		
 supplier_id	supplier_address	supplier_phone
1	Kharkiv, Nauky av., 55, apt. 108	phone: 32-18-44
2	Kyiv, Peremohy av., 154, apt. 3	
3	Kharkiv, Pushkinska str., 77	phone: 33-33-44, fax
4	Odesa, Derebasivska str., 75	
5	Poltava, Soborna str., 15, apt. 43	

## 3.2 Створення запитів на вибірку / Creating select queries

- можна отримувати значення, обчислені за допомогою виразів / it is possible to get calculated values

SELECT supplied\_product, supplied\_amount, supplied\_cost / 27 FROM supplied;

supplied (3×17)		
supplied_product	supplied_amount	supplied_cost / 27
Audio Player	25	25.925926
TV	10	48.148148
Video Player	12	27.777778
Audio Player	5	16.666667
Stereo System	11	18.518519
Video Player	8	16.666667
Audio Player	11	20.370370
Monitor	85	20.370370
TV	52	33.333333

## 3.2 Створення запитів на вибірку / Creating select queries

- можна обчислювати значення без звернення до таблиці / it is possible to calculate values without accessing any table

`SELECT 2 * 2;`

- DISTINCT** – виключити повторювані рядки / exclude repeating records

`SELECT supplied_product | SELECT DISCTINCT supplied_product  
FROM supplied;`

supplied (1×17)	
supplied_product	
Audio Player	
TV	
Video Player	
Audio Player	
Stereo System	
Video Player	
Audio Player	
Monitor	

supplied (1×7)	
supplied_product	
Audio Player	
TV	
Video Player	
Stereo System	
Monitor	
Printer	
Phone	

## 3.2 Створення запитів на вибірку / Creating select queries

- виведені рядки можна впорядкувати по одному з стовпців / selected records can be sorted by one of the columns

**ORDER BY** <column\_name> [**ASC** | **DESC**]

```
SELECT supplied_product,  
       supplied_amount,  
       supplied_cost  
FROM supplied  
ORDER BY  
       supplied_amount DESC,  
       supplied_cost;
```

supplied (3×17)		
supplied_product	supplied_amount	supplied_cost
TV	14	860.00
Video Player	12	750.00
Stereo System	11	500.00
Audio Player	11	550.00
TV	10	1,300.00
TV	10	2,999.00
Video Player	8	450.00
Audio Player	5	450.00
Phone	5	5,999.00

## 3.2 Створення запитів на вибірку / Creating select queries

- замість імен стовпців можна використовувати їх порядкові номери / ordinal numbers of columns can be used instead of their names

SELECT supplied\_product,

supplied\_amount,

supplied\_cost

FROM supplied

ORDER BY 2 DESC, 3;

supplied (3×17)		
supplied_product	supplied_amount	supplied_cost
TV	14	860.00
Video Player	12	750.00
Stereo System	11	500.00
Audio Player	11	550.00
TV	10	1,300.00
TV	10	2,999.00
Video Player	8	450.00
Audio Player	5	450.00
Phone	5	5,999.00

## 3.2 Створення запитів на вибірку / Creating select queries

- об'єднання таблиць / join tables

**SELECT** <columns\_list> **FROM** <tables\_list>

**WHERE** <condition>;

```
SELECT contract_note, supplied_product, supplied_amount, supplied_cost  
FROM supplied, contract
```

```
WHERE contract.contract_number = supplied.contract_number;
```

Result #1 (4×17)			
contract_note	 supplied_product	supplied_amount	supplied_cost
Order 34 on 30.08.2018	Audio Player	25	700.00
Order 34 on 30.08.2018	TV	10	1,300.00
Order 34 on 30.08.2018	Video Player	12	750.00
Invoice 08-78 on 28.08.2018	Audio Player	5	450.00
Invoice 08-78 on 28.08.2018	Stereo System	11	500.00
Invoice 08-78 on 28.08.2018	Video Player	8	450.00
Order 56 on 28.08.2018	Audio Player	11	550.00
Order 56 on 28.08.2018	Monitor	85	550.00
Order 56 on 28.08.2018	TV	52	900.00

## 3.2 Створення запитів на вибірку / Creating select queries

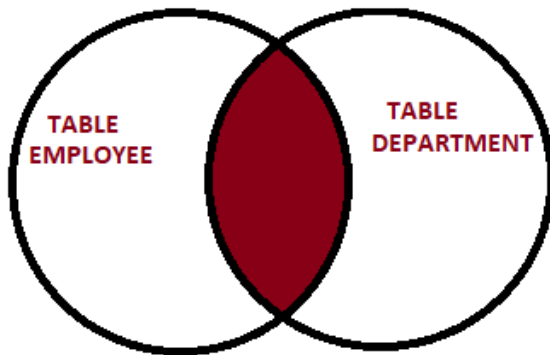
- об'єднання таблиць за допомогою USING / join tables with USING  
**SELECT** <columns\_list> **FROM** <table1> **JOIN** <table2> **JOIN** ...  
**USING** (<table1\_id>, <table2\_id>, ...);

```
SELECT contract_note, supplied_product, supplied_amount, supplied_cost  
FROM supplied INNER JOIN contract  
USING (contract_number);
```

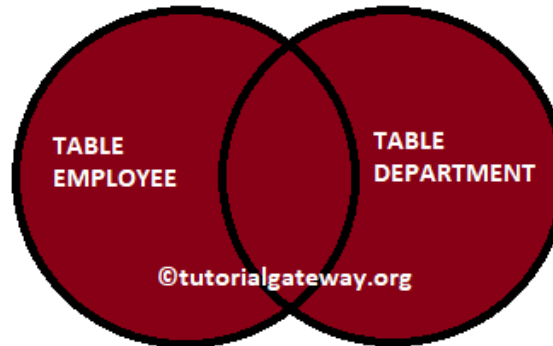
Result #1 (4×17)			
contract_note	 supplied_product	supplied_amount	supplied_cost
Order 34 on 30.08.2018	Audio Player	25	700.00
Order 34 on 30.08.2018	TV	10	1,300.00
Order 34 on 30.08.2018	Video Player	12	750.00
Invoice 08-78 on 28.08.2018	Audio Player	5	450.00
Invoice 08-78 on 28.08.2018	Stereo System	11	500.00
Invoice 08-78 on 28.08.2018	Video Player	8	450.00
Order 56 on 28.08.2018	Audio Player	11	550.00
Order 56 on 28.08.2018	Monitor	85	550.00
Order 56 on 28.08.2018	TV	52	900.00

## 3.2 Створення запитів на вибірку / Creating select queries

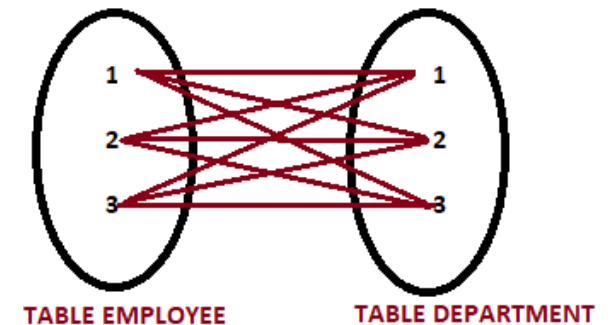
INNER JOIN EXAMPLE



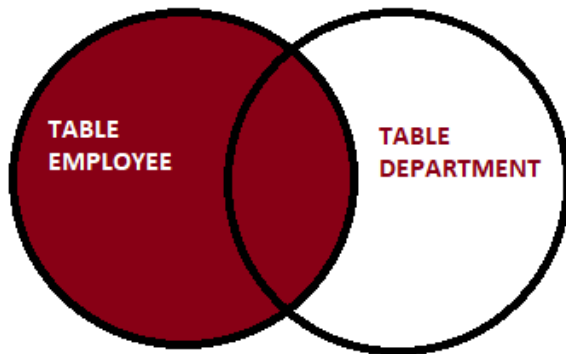
FULL JOIN EXAMPLE



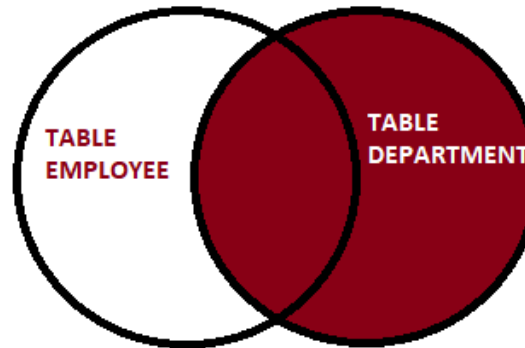
CROSS JOIN EXAMPLE



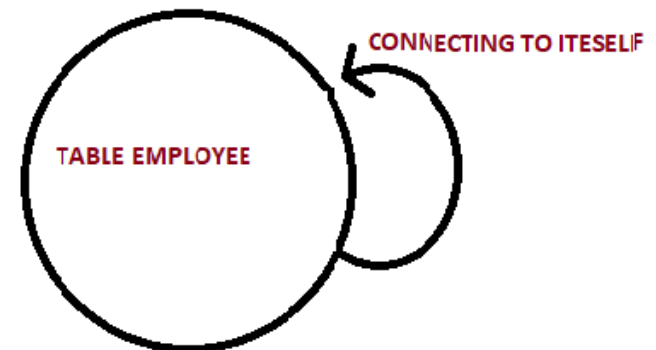
LEFT JOIN EXAMPLE



RIGHT JOIN EXAMPLE



SELF JOIN EXAMPLE





## 3.2 Створення запитів на вибірку / Creating select queries

- таблицю можна об'єднати саму з собою / a table might be join with itself

```
SELECT L.supplied_product,  
       R.supplied_product  
FROM supplied L,  
       supplied R  
WHERE  
       L.supplied_cost =  
       R.supplied_cost
```

supplied (2×21)	
supplied_product	supplied_product
Audio Player	Audio Player
TV	TV
Video Player	Video Player
Audio Player	Audio Player
Video Player	Audio Player
Stereo System	Stereo System
Audio Player	Video Player
Video Player	Video Player
Audio Player	Audio Player

## 3.2 Створення запитів на вибірку / Creating select queries

Додаткова умова відбору, щоб позбутися від повторень

Additional filtering condition to avoid repeating values

```
SELECT L.supplied_product,  
       R.supplied_product
```

```
FROM supplied L,  
      supplied R
```

```
WHERE
```

```
L.supplied_cost = R.supplied_cost AND
```

```
L.supplied_product <> R.supplied_product
```

supplied (2×4)	
supplied_product	supplied_product
Video Player	Audio Player
Audio Player	Video Player
Monitor	Audio Player
Audio Player	Monitor

## 3.2 Створення запитів на вибірку / Creating select queries

- результат одного запиту можна використовувати в іншому запиті / a query result can be used in another query

```
SELECT * FROM supplied
```

```
WHERE supplied_amount = (SELECT MAX(supplied_amount)  
FROM supplied)
```

supplied (4×1)		supplied (4×1)	
contract_number	supplied_product	supplied_amount	supplied_cost
3	Monitor	85	550.00

```
SELECT * FROM supplied
```

```
ORDER BY supplied_amount DESC
```

```
LIMIT 1
```

supplied (4×1)		supplied (4×1)	
contract_number	supplied_product	supplied_amount	supplied_cost
3	Monitor	85	550.00

## 3.2 Створення запитів на вибірку / Creating select queries

- результати декількох запитів можна об'єднати / results of several queries can be combined

```
SELECT * FROM supplied
```

```
WHERE supplied_amount = (SELECT MAX(supplied_amount) FROM supplied)
```

**UNION**

```
SELECT * FROM supplied
```

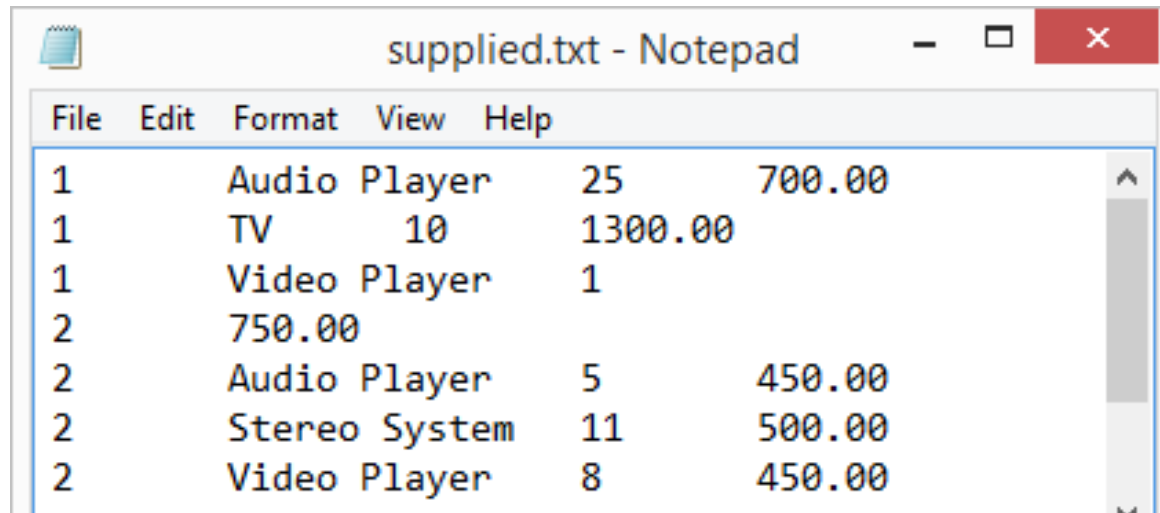
```
WHERE supplied_amount = (SELECT MIN(supplied_amount) FROM supplied)
```

Result #1 (4×3)			
contract_number	supplied_product	supplied_amount	supplied_cost
3	Monitor	85	550.00
2	Audio Player	5	450.00
7	Phone	5	5,999.00

## 3.2 Створення запитів на вибірку / Creating select queries

- результат запиту можна зберегти в файл / a query result can be saved to a file

**SELECT \* FROM supplied INTO OUTFILE**  
**'D:/supplied.txt'**



1	Audio Player	25	700.00
1	TV	10	1300.00
1	Video Player	1	750.00
2	Audio Player	5	450.00
2	Stereo System	11	500.00
2	Video Player	8	450.00

## 3.3 Умови вибірки / Conditions


### WHERE

вибір записів, які відповідають певним критеріям пошуку

select records that satisfy specific search criterias


book.users: 6 rows total (approximately)

Next

 user_ID	u_name	u_patronymic	u_surname	u_phone	u_email	u_status
1	Александр	Валерьевич	Иванов	58-98-78	ivanov@email.ru	active
2	Сергей	Иванович	Лосев	90-57-77	losev@email.ru	passive
3	Игорь	Николаевич	Симонов	95-66-61	simonov@email.ru	active
4	Максим	Петрович	Кузнецов	(NULL)	kuznetsov@email.ru	active
5	Анатолий	Юрьевич	Петров	(NULL)	(NULL)	lock
6	Александр	Александрович	Корнеев	89-78-36	korneev@email.ru	gold

## 3.3 УМОВИ вибірки / Conditions

```
SELECT user_ID, u_surname FROM users  
WHERE u_status = 'active';
```

users (2×3)	
 user_ID	u_surname
1	Іванов
3	Симонов
4	Кузнецов

```
SELECT DISTINCT u_status FROM users;
```

users (1×4)	
u_status	
active	
passive	
lock	
gold	

## 3.4 Сортвання / Sorting

```
SELECT * FROM orders ORDER BY o_user_ID;
```

orders (5×5)				
order_ID	o_user_ID	o_book_ID	o_time	o_number
3	1	20	2009-02-18 13:41:05	4
1	3	8	2009-01-04 10:39:38	1
5	3	20	2009-03-17 19:15:36	1
4	4	20	2009-03-10 18:20:00	1
2	6	10	2009-02-10 09:40:29	2

```
SELECT o_time FROM orders ORDER BY o_time DESC;
```

orders (1×5)	
o_time	
2009-03-17 19:15:36	
2009-03-10 18:20:00	
2009-02-18 13:41:05	
2009-02-10 09:40:29	
2009-01-04 10:39:38	



## 3.4 Сортювання / Sorting


- результат вибірки – записи в порядку зберігання в БД / select result – records in an order according to which they are stored in a DB
- можна сортювати за кількома стовпцями / it is possible to sort by multiple columns
- за замовчуванням сортювання виконується в прямому порядку / records are sorted in an ascending order by default (**ASC**)
- можна впорядкувати рядки в зворотному порядку / it is possible to sort records in a descending order (**DESC**)

## 3.5 Обмеження вибірки / Limit

### LIMIT

використовується для реалізації посторінкової навігації  
is used to implement navigation by pages (pagination)

```
SELECT book_ID, b_count FROM books  
ORDER BY b_count DESC  
LIMIT 5;
```

books (2×5)	
 book_ID	b_count
28	20
25	20
26	15
29	12
9	12


## 3.5 Обмеження вибірки / Limit

### LIMIT offset, count

**offset** – позиція, починаючи з якої необхідно повернути результат / the offset of the first row to be returned

**count** – число видобутих записів / the maximum number of rows to be returned

```
SELECT book_ID, b_count FROM books  
ORDER BY b_count DESC  
LIMIT 5, 5;
```

books (2×5)	
 book_ID	b_count
1	10
27	10
24	8
30	8
19	6

## 3.5 Обмеження вибірки / Limit

books (3×30)		
book_ID	b_name	b_price
1	JavaScript в кармане	42.00
2	Visual FoxPro 9.0	660.00
3	C++ Как он есть	218.00
4	Создание приложений с помощью C#	169.00
5	Delphi. Народные советы	243.00
6	Delphi. Полное руководство	500.00
7	Профессиональное программирование на PHP	309.00
8	Совершенный код	771.00
9	Практика программирования	214.00
10	Принципы маршрутизации в Internet	428.00
11	Поиск в Internet	107.00
12	Web-конструирование	177.00
13	Самоучитель Интернет	121.00
14	Популярные интернет-браузеры	82.00
15	Общение в Интернете	85.00
16	Базы данных	326.00
17	Базы данных. Разработка приложений	189.00
18	Раскрытие тайн SQL	200.00
19	Практикум по Access	87.00
20	Компьютерные сети	630.00

SELECT book\_ID,  
b\_name, b\_price  
FROM books  
LIMIT 0, 5

SELECT book\_ID,  
b\_name, b\_price  
FROM books  
LIMIT 5, 5

SELECT book\_ID,  
b\_name, b\_price  
FROM books  
LIMIT 10, 5

SELECT book\_ID,  
b\_name, b\_price  
FROM books  
LIMIT 15, 5