

## 2.3 Створення таблиць і робота зі структурою БД / Creating tables and working with the DB structure

**CREATE [TEMPORARY] TABLE [IF NOT EXISTS] ( ... );**

**TEMPORARY**

створення таблиці, яка існує тільки в поточному сеансі роботи з БД

creates the table that exists only in the current session

**IF NOT EXISTS**

таблиця буде створена, якщо ще немає таблиці з вказаним ім'ям

the table will be created if there is no table with the specified name

## 2.3 Створення таблиць і робота зі структурою БД / Creating tables and working with the DB structure

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS]  
<table name> LIKE <existing table name>
```

Створення таблиці з такою ж структурою, як у існуючої таблиці

Creating the table with the same structure the existing table has

## 2.3 Створення таблиць і робота зі структурою БД / Creating tables and working with the DB structure

- Індексація стовпчика / Column indexing  
INDEX, KEY

стовпці не обов'язково містять унікальні значення

columns are not necessary contain unique values

UNIQUE

стовпці повинні містити унікальні значення

columns should contain unique values

## 2.3 Створення таблиць і робота зі структурою БД / Creating tables and working with the DB structure

- Індиксація стовпчика / Column indexing

### FULLTEXT

повнотекстові індекси для стовпців типу TEXT, CHAR, VARCHAR

full text indexes based on columns of types TEXT, CHAR, and VARCHAR

! Тільки для таблиць типу **MyISAM** tables only!

## 2.3 Створення таблиць і робота зі структурою БД / Creating tables and working with the DB structure

- Індокси - основний спосіб прискорення роботи БД
- Using indexes is the main way to improve the database performance
- Для стовпця створюється копія, постійно підтримувана в відсортованому стані
- Indexed column is copied and maintained in sorted state constantly

### **PRIMARY KEY, KEY, UNIQUE, INDEX**

CREATE INDEX <name> ON <table>(<field>);

## 2.3 Створення таблиць і робота зі структурою БД / Creating tables and working with the DB structure

DESCRIBE <table name>;

Дозволяє отримати детальну інформацію про структуру таблиці

Provides the detailed information about the table structure

COLUMNS (6×4)					
Field	Type	Null	Key	Default	Extra
contract_number	int(11)	NO	PRI	(NULL)	auto_increment
contract_date	timestamp	NO		CURRENT_TIMESTAMP	on update CURRENT_TIMESTAMP
supplier_id	int(11)	NO	MUL	(NULL)	
contract_note	varchar(100)	YES		(NULL)	

## 2.3 Створення таблиць і робота зі структурою БД / Creating tables and working with the DB structure

### Типи даних для стовпців / Column data types

- Числові / Numeric
- Строкові / String
- Календарні / Calendar
- NULL – позначає відсутність інформації / identifies absence of information

## 2.3 Створення таблиць і робота зі структурою БД / Creating tables and working with the DB structure

### Числові типи / Numeric types

- Точні типи / Exact types

**INTEGER** та його варіації / and its variations

**DECIMAL**

- Приблизні типи / Approximate types

**FLOAT**

**DOUBLE**



## 2.3 Створення таблиць і робота зі структурою БД / Creating tables and working with the DB structure

### **DECIMAL (M, D)**

M – число символів для відображення всього числа / number of digits that represent total number

D – число символів для відображення дробової частини / number of digits in its fractional part

price DECIMAL (5, 2)

## 2.3 Створення таблиць і робота зі структурою БД / Creating tables and working with the DB structure

### **UNSIGNED**

стовпець містить тільки позитивні числа або нулі  
a column stores only positive numbers or zero values

### **ZEROFILL**

число буде відображатися з провідними нулями  
a number will be shown with leading zero digits

## 2.3 Створення таблиць і робота зі структурою БД / Creating tables and working with the DB structure

### Текстові типи і рядки / Text and string types

- CHAR – зберігання рядків фіксованої довжини / stores strings of the fixed length
- VARCHAR – зберігання рядків змінної довжини / stores strings of the variable length
- TEXT, BLOB – зберігання великих фрагментів тексту / stores large fragments of text
- ENUM, SET – зберігання значень із заданого списку / stores values from a pre-defined list

## 2.3 Створення таблиць і робота зі структурою БД / Creating tables and working with the DB structure

### **CHAR(M)**

незалежно від довжини рядка, використовує для її зберігання всіх M символів

does not depend on the string length, uses all M chars to store it

### **VARCHAR (M)**

використовує кількість символів, яка дорівнює довжині рядка + 1 байт

uses the number of chars which is equal to the string length + 1 byte

## 2.3 Створення таблиць і робота зі структурою БД / Creating tables and working with the DB structure

- CHAR обробляється ефективніше VARCHAR
- CHAR is more efficient than VARCHAR
- Не можна змішувати в таблиці стовпці CHAR і VARCHAR
- Do not mix CHAR and VARCHAR columns in a table

CHAR => VARCHAR

## 2.3 Створення таблиць і робота зі структурою БД / Creating tables and working with the DB structure

### TEXT

поддерживает полнотекстовый поиск  
supports full-text search

TEXT	BLOB
Враховується кодування Character set is taken into account	Не враховується кодування Character set is not taken into account
Для зберігання великих обсягів тексту Used to store large amounts of text	Для зберігання великих двійкових об'єктів Used to store large binary objects

## 2.3 Створення таблиць і робота зі структурою БД / Creating tables and working with the DB structure

### **ENUM**

містить одне значення із зазначеної множини  
contains a single value from the pre-defined set

### **SET**

може містити будь-який або всі елементи  
заданої множини одночасно

might contain any or all elements from the pre-defined set at the same time

status ENUM ('valid', 'invalid') **DEFAULT** 'invalid'

## 2.3 Створення таблиць і робота зі структурою БД / Creating tables and working with the DB structure

Календарні типи даних / Calendar data types

**DATE** (YYYY-MM-DD)

**TIME** (HH:MM:SS)

**DATETIME** (YYYY-MM-DD HH:MM:SS)

**TIMESTAMP**

**YEAR** (YYYY)

В якості роздільників можуть виступати будь-які символи, відмінні від цифри

Any symbols might be used as separators, except digits



## 2.3 Створення таблиць і робота зі структурою БД / Creating tables and working with the DB structure

- Обробка числових даних відбувається швидше строкових
- Numeric data is processed faster than string data
- Продуктивність можна збільшити за рахунок подання рядків у вигляді чисел
- Performance might be improved by representing string values as numeric values

## 2.3 Створення таблиць і робота зі структурою БД / Creating tables and working with the DB structure

- Якщо є можливість, слід вибирати типи даних, що займають менше місця
- Choose data types that require less storage space if it is possible
- Типи фіксованої довжини обробляються швидше типів змінної довжини
- Data types with fixed length are processed faster than data types with variable length
- Інакше необхідно використовувати **OPTIMIZE TABLE** для дефрагментації таблиці
- Otherwise use **OPTIMIZE TABLE** to defragment the table

## 2.3 Створення таблиць і робота зі структурою БД / Creating tables and working with the DB structure

- [CONSTRAINT <key\_name>] **PRIMARY KEY** (<columns\_list>)
- **INDEX** [<index\_name>] (<columns\_list>)
- [CONSTRAINT <index\_name>] **UNIQUE** [index\_name] (<columns\_list>)

### TEXT, BLOB

вказати кількість символів на початку значення, за якими буде проведено індексування

specify the number of symbols at the beginning of a value that will be used for indexing

## 2.3 Створення таблиць і робота зі структурою БД / Creating tables and working with the DB structure

**FULLTEXT** [<index\_name>] (<columns\_list>)

прискорений пошук за значеннями символьних стовпців

fast search over values of character columns

**CHAR, VARCHAR, TEXT**

можна створити тільки в таблицях типу MyISAM  
can be created only in MyISAM tables

## 2.3 Створення таблиць і робота зі структурою БД / Creating tables and working with the DB structure

```
[CONSTRAINT <fk_name>] FOREIGN KEY  
[<index_name>] (<columns_list>)  
REFERENCES KEY [<parent_table_name>]  
(<pk_columns_list>) [<integrity_rules>]
```

тільки для таблиць з типом InnoDB (і дочірня і батьківська повинні мати тип **InnoDB**)

only for InnoDB tables (both child and parent tables should be **InnoDB** tables)

## 2.3 Створення таблиць і робота зі структурою БД / Creating tables and working with the DB structure

- для числових стовпців повинен збігатися розмір і знак
- sizes and signs should match for numeric columns
- для символічних стовпців повинне співпадати кодування і правило порівняння значень
- character sets and collations should match for character columns
- стовпчики з типом TEXT і BLOB не можуть входити в зовнішній ключ
- TEXT and BLOB columns can not be included into a foreign key

## 2.3 Створення таблиць і робота зі структурою БД / Creating tables and working with the DB structure

### Опціональні властивості таблиці / Optional table properties

- ENGINE <table\_type>
- AUTO\_INCREMENT <start\_value>
- CHARACTER SET <charset\_name>
- COLLATE <comparison\_rule>

## 2.3 Створення таблиць і робота зі структурою БД / Creating tables and working with the DB structure

### **ALTER TABLE**

використовується для модифікації раніше створеної таблиці

used to modify previously created table

**ALTER TABLE** <table\_name>

**ADD** <column\_name> <column\_type>  
[<column\_properties>]

[**FIRST** | **AFTER** <preceding\_column\_name>];



## 2.3 Створення таблиць і робота зі структурою БД / Creating tables and working with the DB structure

### **FIRST, AFTER**

дозволяють вказати місце доданого стовпця  
allow to specify the place of the added column

- **FIRST** – стовпець може стати першим / a column could be the first
- **AFTER** – може слідувати за вказаними стовпцем / could follow the specified preceding column
- Стає останнім, якщо місце не вказано / It would be the last column if the place is not specified

## 2.3 Створення таблиць і робота зі структурою БД / Creating tables and working with the DB structure

**ALTER TABLE** <table\_name> **ADD [CONSTRAINT**  
    <key\_name>] **PRIMARY KEY** (<columns\_list>);

стовпці, які будуть входити в первинний ключ,  
повинні вже існувати в таблиці

columns included into the PK, should already exist  
in a table

```
ALTER TABLE supplier ADD PRIMARY KEY  
    (supplier_id)
```

## 2.3 Створення таблиць і робота зі структурою БД / Creating tables and working with the DB structure

- Додавання зовнішнього ключа / Add a foreign key

```
ALTER TABLE <table_name> ADD [CONSTRAINT  
  <fk_name>] FOREIGN KEY [<index_name>]  
  (<columns_list>) REFERENCES KEY  
  [<parent_table_name>] (<pk_columns_list>)  
  [<integrity_rules>]
```

- Додавання індекса / Add index

```
ALTER TABLE <table_name> ADD INDEX  
  [<index_name>] (<columns_list>);
```

## 2.3 Створення таблиць і робота зі структурою БД / Creating tables and working with the DB structure

- Додавання унікального індексу / Add unique index

```
ALTER TABLE <table_name> ADD [CONSTRAINT  
    <constraint_name>] UNIQUE (<columns_list>);
```

- Додавання повнотекстового індексу / Add full text index

```
ALTER TABLE <table_name> ADD FULLTEXT  
    [<index_name>] (<columns_list>);
```

## 2.3 Створення таблиць і робота зі структурою БД / Creating tables and working with the DB structure

- Зміна опису стовпчика / Change column description

**ALTER TABLE** <table\_name> **CHANGE** <old\_name>  
<new\_name> <new\_type> [<column\_properties>]  
[**FIRST** | **AFTER** <preceding\_column\_name>];

- Зміна опису стовпчика без перейменування /  
Change column description without rename

**ALTER TABLE** <table\_name> **MODIFY** <column\_name>  
<new\_type> [<column\_properties>] [**FIRST** | **AFTER**  
<preceding\_column\_name>];

## 2.3 Створення таблиць і робота зі структурою БД / Creating tables and working with the DB structure

### Значення за замовчуванням / Default values

- Встановити значення за замовчуванням / Set the default value

**ALTER TABLE <table\_name> ALTER <column\_name>  
SET DEFAULT <default\_value>;**

- Видалити значення за замовчуванням / Delete the default value

**ALTER TABLE <table\_name> ALTER <column\_name>  
DROP DEFAULT;**

## 2.3 Створення таблиць і робота зі структурою БД / Creating tables and working with the DB structure

Видалення стовпця / Delete column

**ALTER TABLE** <table\_name> **DROP** <column\_name>;

- стовпець вилучено з усіх індексів / column is deleted from all indexes
- первинний і зовнішні ключі потрібно видалити перш, ніж видаляти включені до них стовпці / PK and FK should be removed before included columns would be removed
- порожній індекс видаляється автоматично / empty index is removed automatically

## 2.3 Створення таблиць і робота зі структурою БД / Creating tables and working with the DB structure

**ALTER TABLE <table\_name> DROP PRIMARY KEY;**

**ALTER TABLE <table\_name> DROP FOREIGN KEY <fk\_name>;**

ім'я зовнішнього ключа, якщо не було вказано, генерується  
автоматично

if the FK name was not specified it would be generated  
automatically

**ALTER TABLE <table\_name> DROP INDEX <index\_name>;**

**SHOW CREATE TABLE <table\_name>;**



## 2.3 Створення таблиць і робота зі структурою БД / Creating tables and working with the DB structure

- Перейменування таблиці / Rename table

**ALTER TABLE** <table\_name> **RENAME**  
<new\_table\_name>;

- Зміна кодування і правила порівняння /  
Change the character set and collation

**ALTER TABLE** <table\_name> **CHARACTER SET**  
<charset\_name> [**COLLATE**  
<collation\_name>];

## 2.3 Створення таблиць і робота зі структурою БД / Creating tables and working with the DB structure

- Перетворити в нову систему кодування існуючі стовпчики / Convert existing columns to the new character set

**ALTER TABLE** <table\_name> **CONVERT TO CHARACTER SET** <charset\_name> [**COLLATE** <collation\_name>];

- Видалення таблиці / Delete a table
- DROP TABLE** <table\_name>;

## 2.3 Створення таблиць і робота зі структурою БД / Creating tables and working with the DB structure

Основні операції з таблицями

Basic operations on tables

- CREATE TABLE
- ALTER TABLE
- DROP TABLE
- SHOW TABLES
- DESCRIBE
- SHOW CREATE TABLE

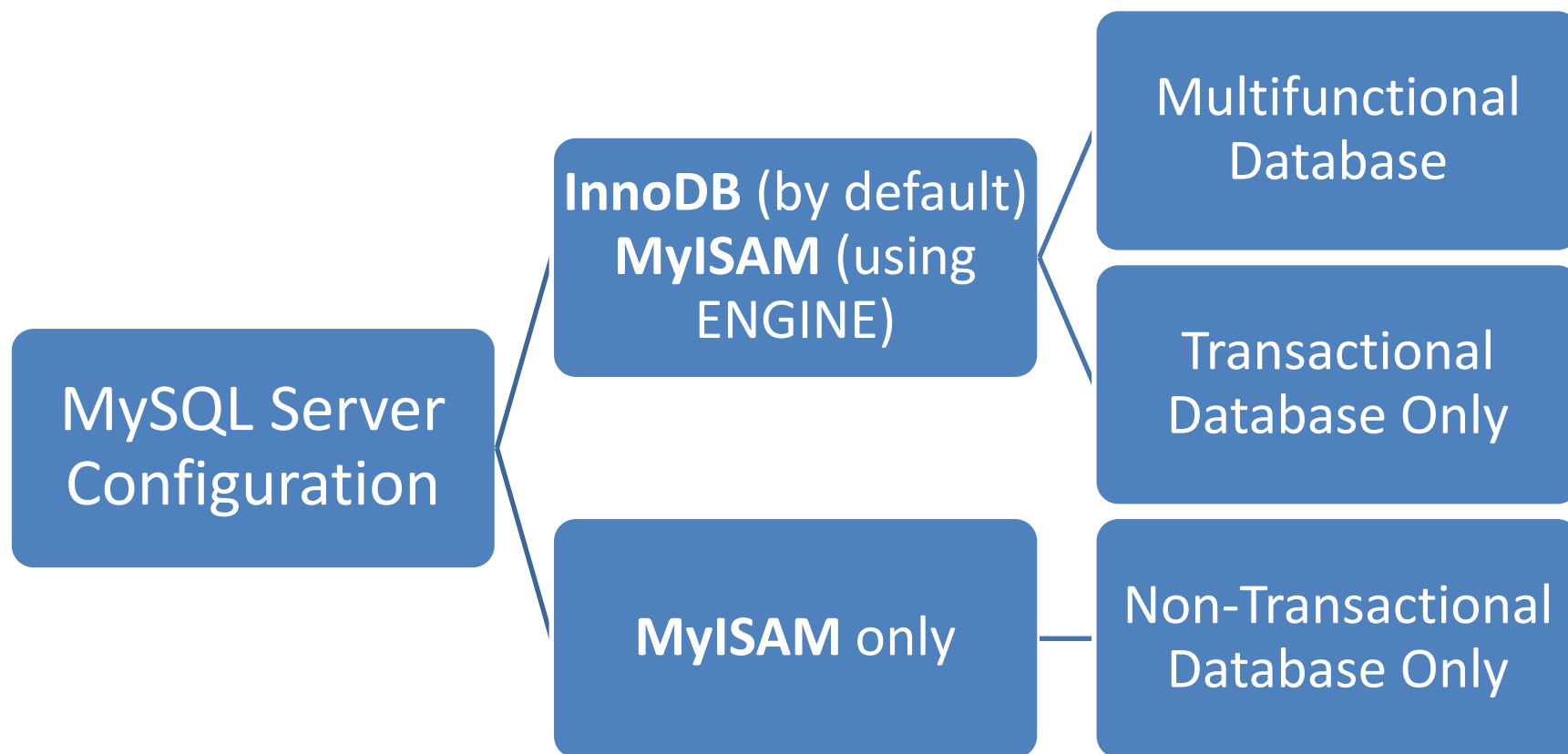
## 2.4 Типи таблиць в MySQL / Table types in MySQL

**ENGINE** <table\_type>

Основні типи таблиць / Main table types

- **InnoDB** – підтримує транзакції, зовнішні ключі, каскадне видалення і блокування на рівні рядків / supports transactions, cascade delete, and row-based locks
- **MyISAM** – дуже швидка робота, підтримує повнотекстову індексацію / extremely high performance, supports full text indexing

## 2.4 Типы таблиц в MySQL / Table types in MySQL



## 2.4 Типи таблиць в MySQL / Table types in MySQL

- **BDB (Berkeley DB)** – нереляційний механізм, зберігає пари «ключ-значення» / non-relational mechanism used to store “key-value” pairs
- **MEMORY** – таблиці цілком зберігаються в оперативній пам'яті / tables are completely stored in RAM
- **MERGE** – об'єднання декількох таблиць MyISAM з однією структурою / merge several MyISAM tables with the similar structure

## 2.4 Типи таблиць в MySQL / Table types in MySQL

- **NDB Cluster** – розподіл таблиць між декількома комп'ютерами / distribution of tables across multiple computers
- **ARCHIVE** – зберігання великого обсягу даних в стислому форматі, підтримка тільки SELECT і INSERT / stores huge amount of data using a compressed format, supports only SELECT and INSERT
- **CSV** – текстовий файл / text file
- **FEDERATED** – дані в таблицях зберігаються на іншому комп'ютері в мережі / tables data is stored in another computer in a network