

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**

Інститут комп'ютерних наук та інформаційних технологій

Кафедра систем штучного інтелекту



Звіт до лабораторної роботи №2

з дисципліни

“Організація Баз Даних та знань”

Виконала:

ст. гр. КН-210
Заремба Вікторія

Викладач:

Мельникова Н.І.

Лабораторна робота №3
з курсу “ОБДЗ”
на тему:
“Модифікація структури таблиць бази даних засобами SQL”

Мета роботи: Розробити SQL запити для зміни структури таблиць бази даних (включення нових полів, вилучення полів таблиць, зміни опису полів, зміни обмежень).

Короткі теоретичні відомості.

Для зміни структури таблиць використовується наступна директива мови SQL.

Синтаксис:

ALTER [IGNORE] TABLE ім'я_таблиці
опис_зміни [, опис_зміни ...]

опис_зміни:

[додаткові_параметри ...]
| ADD [COLUMN] назва_поля опис_поля
[FIRST | AFTER назва_поля]
| ADD [COLUMN] (назва_поля опис_поля,...)
| ADD {INDEX|KEY} [ім'я_обмеження] (назва_поля,...)[тип_обмеження]
| ADD [CONSTRAINT [ім'я_обмеження]] PRIMARY KEY (назва_поля,...)
[тип_обмеження]
| ADD [CONSTRAINT [ім'я_обмеження]]
UNIQUE [INDEX|KEY] [ім'я_обмеження](назва_поля,...)[тип_обмеження]
| ADD [FULLTEXT|SPATIAL] [INDEX|KEY] [ім'я_обмеження]
(назва_поля,...) [тип_обмеження]
| ADD [CONSTRAINT [ім'я_обмеження]]
FOREIGN KEY [ім'я_обмеження] (назва_поля,...) опис_зв'язку
| ALTER [COLUMN] назва_поля {SET DEFAULT значення | DROP DEFAULT}
| CHANGE [COLUMN] стара_назва_поля нова_назва_поля опис_поля
[FIRST|AFTER назва_поля]
| MODIFY [COLUMN] назва_поля опис_поля
[FIRST | AFTER назва_поля]
| DROP [COLUMN] назва_поля
| DROP PRIMARY KEY
| DROP {INDEX|KEY} ім'я_обмеження
| DROP FOREIGN KEY ім'я_обмеження
| RENAME [TO] нова_назва_таблиці
| ORDER BY назва_поля [,назва_поля] ...
| CONVERT TO CHARACTER SET кодування [COLLATE набір_правил]
| [DEFAULT] CHARACTER SET [=] кодування [COLLATE [=] набір_правил]

додаткові_параметри:

{ENGINE|TYPE} [=] тип_таблиці
| AUTO_INCREMENT [=] початкове_значення
| AVG_ROW_LENGTH [=] значення
| [DEFAULT] CHARACTER SET [=] кодування
| CHECKSUM [=] {0 | 1}
| [DEFAULT] COLLATE [=] набір_правил
| COMMENT [=] 'коментар до таблиці'
| DELAY_KEY_WRITE [=] {0 | 1}

| MAX_ROWS [=] значення
| MIN_ROWS [=] значення
| ROW_FORMAT {DEFAULT|DYNAMIC|FIXED|COMPRESSED|REDUNDANT|COMPACT}

Аргументи:

тип_обмеження

Вказує тип індексу для поля: USING {BTREE | HASH | RTREE}.

IGNORE

Ігнорувати помилки дублювання значень в ключових полях, тобто залишати в таблиці перше значення, і не вставляти дублікат. За замовчуванням, при такій помилці вся дія над таблицею відміняється.

ALTER назва_поля

Використовується для зміни значення за замовчуванням для поля.

MODIFY назва_поля

Змінити тип поля.

CHANGE назва_поля

Змінити назву і тип поля.

FIRST | AFTER

Змінити розташування поля в таблиці. За замовчуванням, нове поле додається в кінець таблиці. FIRST додасть його на початок; AFTER – одразу після вказаного поля.

DROP назва_поля

Видалення поля, обмеження чи індексу.

ORDER BY

Відсортувати рядки в таблиці за зростанням значень у вказаному полі (полях).

AUTO_INCREMENT

При зміні початкового значення для приросту поля-лічильника, слід вказати значення більше за ті, які вже містяться в цьому полі.

CONVERT TO

Задає кодування, в яке потрібно перевести таблицю.

RENAME TO

Змінити назву таблиці на вказану. Для зміни назви таблиці, або для переносу таблиці в іншу базу, також можна використати директиву RENAME TABLE ім'я_бази_1.назва_таблиці_1 TO ім'я_бази_2.назва_таблиці_2.

Хід Роботи

Виконаємо команди: **SHOW TABLES; DESCRIBE budget_investor.**

Tables_in_pm_system	
1	budget
2	budget_investor
3	department
4	investor
5	project
6	task
7	task_user_department
8	user
9	user_department

	Field	Type	Null	Key	Default	Extra
1	id_project	int unsigned	NO	PRI	<null>	auto_increment
2	name	varchar(99)	NO		<null>	
3	date	date	NO		<null>	
4	active	tinyint(1)	NO		<null>	
5	purpose	text	YES		<null>	
6	place	text	YES		<null>	
7	target_audience	varchar(100)	YES		<null>	
8	comments	text	YES		<null>	

Виконаємо такі операції над схемою бази даних:

1. Вилучимо зв'язок `fk_tud_ud`
2. Вилучимо і додамо поля в таблиці `task`
3. Змінимо поле в таблиці `User`
4. Додамо поле і нове обмеження унікальності до таблиці `task`
5. Змінимо тип обмеження цілісності для зв'язків `fk_department_project`, `fk_task_department`, `fk_user_user_department`, `fk_department_user_department`, `fk_tud_ud`, `fk_tud_task`

1. Вилучимо зв'язок `fk_tud_ud`

Видалення зв'язку між таблицями шляхом видалення обмеження для таблиці:

```
ALTER TABLE pm_system.task_user_department
    DROP FOREIGN KEY fk_tud_ud,
    DROP INDEX fk_tud_ud;
```

Для перевірки результату можна виконати команди:

- `SHOW INDEX FROM pm_system.task_user_department`

	Table	Non_unique	Key_name	Seq_in_index	Column_name	Collat
1	task_user_department	0	PRIMARY		1 id_task_user_department	A
2	task_user_department	1	fk_tud_task		1 id_task	A

- `SHOW CREATE TABLE pm_system.task_user_department.`

```
CREATE TABLE `task_user_department` (
  `id_task_user_department` int unsigned NOT NULL AUTO_INCREMENT,
  `id_user_department` int unsigned NOT NULL,
  `id_task` int unsigned NOT NULL,
  `start_data` date NOT NULL,
  PRIMARY KEY (`id_task_user_department`),
  KEY `fk_tud_task` (`id_task`),
```

```
CONSTRAINT `fk_tud_task` FOREIGN KEY (`id_task`) REFERENCES `task` (`id_task`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
```

(після цього повертаємо видалений зв'язок)

```
ALTER TABLE pm_system.task_user_department
    ADD CONSTRAINT fk_tud_ud FOREIGN KEY(id_user_department)
    REFERENCES PM_System.user_department(id_user_department) ON UPDATE NO ACTION ON
DELETE NO ACTION;
```

2. Вилучимо і додамо поля в таблиці task

Вилучимо поле "status" і додамо поле "start"

```
ALTER TABLE pm_system.task
    DROP COLUMN status,
    ADD COLUMN start DATE;
```

3. Змінімо поле в таблиці User

Змінюємо поле phone_number: зберігаємо розмірність, але робимо його необов'язковим

```
ALTER TABLE pm_system.user
    MODIFY COLUMN phone_number char(12) NULL;
```

4. Додамо поле і нове обмеження унікальності до таблиці task

Змінюємо тип поля priority на ENUM ('low', 'high', 'medium') та задаємо дефолтне значення DEFAULT 'medium'.

Додаємо поле id_version і створюємо нове обмеження унікальності для полів (id_task, id_version)

```
ALTER TABLE pm_system.task
    MODIFY priority ENUM ('low', 'high', 'medium') DEFAULT 'medium',
    ADD COLUMN id_version TINYINT(8) UNSIGNED NOT NULL AFTER id_task,
    ADD CONSTRAINT UNIQUE OQ_task_version (id_task, id_version);
```

Переглянемо результат за допомогою команди DESCRIBE pm_system.task;

Field	Type	Null	Key	Default	Extra
1 id_task	int unsigned	NO	PRI	<null>	auto_increment
2 id_version	tinyint uns...	NO		<null>	
3 name	text	NO		<null>	
4 details	text	YES		<null>	
5 deadline	date	YES		<null>	
6 comment	text	YES		<null>	
7 priority	enum('low',...	YES		medium	
8 id_department	int unsigned	NO	MUL	<null>	
9 start	date	YES		<null>	
10 status	tinyint(1)	YES		<null>	

5. Змінімо тип обмеження цілісності для зв'язків fk_department_project, fk task department, fk user user department, fk department user department, fk tud ud, fk tud task

Змінімо обмеження цілісності для цих зв'язків, щоб при видаленні проектів з бази видалялись всі його складові (департаменти, завдання...). Для цього потрібно видалити існуюче обмеження і створити нове.

- *fk_department_project*

```
ALTER TABLE pm_system.department
    DROP FOREIGN KEY fk_department_project;
ALTER TABLE pm_system.department
    ADD CONSTRAINT fk_department_project FOREIGN KEY (id_project)
    REFERENCES pm_system.project (id_project) ON DELETE CASCADE;
```

- *fk_task_department*

```
ALTER TABLE pm_system.task
    DROP FOREIGN KEY fk_task_department;
ALTER TABLE pm_system.task
    ADD CONSTRAINT fk_task_department FOREIGN KEY (id_department)
    REFERENCES pm_system.department (id_department) ON DELETE CASCADE;
```

- *fk_user_user_department*

```
ALTER TABLE pm_system.user_department
    DROP FOREIGN KEY fk_user_user_department;
ALTER TABLE pm_system.user_department
    ADD CONSTRAINT fk_user_user_department FOREIGN KEY(id_user)
    REFERENCES PM_System.user(id_user) ON DELETE CASCADE;
```

- *fk_department_user_department*

```
ALTER TABLE pm_system.user_department
    DROP FOREIGN KEY fk_department_user_department;
ALTER TABLE pm_system.user_department
    ADD CONSTRAINT fk_department_user_department FOREIGN KEY(id_department)
    REFERENCES PM_System.department(id_department) ON DELETE CASCADE;
```

- *fk_tud_ud*

```
ALTER TABLE pm_system.task_user_department
    DROP FOREIGN KEY fk_tud_ud;
ALTER TABLE pm_system.task_user_department
    ADD CONSTRAINT fk_tud_ud FOREIGN KEY(id_user_department)
    REFERENCES PM_System.user_department(id_user_department) ON DELETE
CASCADE;
```

- *fk_tud_task*

```
ALTER TABLE pm_system.task_user_department
    DROP FOREIGN KEY fk_tud_task;
ALTER TABLE pm_system.task_user_department
    ADD CONSTRAINT fk_tud_task FOREIGN KEY(id_task)
    REFERENCES PM_System.task(id_task) ON DELETE CASCADE;
```

Висновок: на цій лабораторній роботі було розглянуто методи модифікації структури таблиць БД засобами SQL та проведено модифікацію полів, зв'язків і обмежень