Бази даних

Лекція 4

Тематика лекції

- SQL - частина 1

SQL

- **DDL - Data Definition Language - маніпуляції схемою бази даних, наприклад створення таблиць і тд.**
- **DML - Data Manipulation Language - операції з даними, що зберігаються в базі даних.**
- **DCL - Data Control Language - контроль доступів до даних.**
- **TCL - Transaction Control Language - контроль транзакційних операцій.**

DML

- **INSERT - додавання нових рядків до бази даних.**
- **SELECT - читання даних з бази даних.**
- **UPDATE - зміна вже записаних даних.**
- **DELETE - видалення даних.**

INSERT

INSERT INTO <table name> [(<column 1>, [column 2], ...)]
VALUES (<value 1>, [value 2], ...);

INSERT INTO student (name, surname, profession, contact_data_id, group_id)
VALUES ('Андрій', 'Ковальчук', 121, 11, 1);

SELECT

```
SELECT <column 1>, [column 2], ...
FROM <table name>
WHERE <conditions>;
```

```
SELECT *
FROM student
WHERE name = 'Андрій' AND  surname = 'Ковальчук';
```

SELECT with alias

```
SELECT *
FROM student;
```

```
SELECT
    student_id AS “student id”,
    name,
    surname
FROM student;
```

SELECT з обробкою даних

```
SELECT
    student_id AS id,
    concat(name, ' ', surname) AS full_name
FROM student;
```

SELECT DISTINCT

```
SELECT
    profession
FROM student;
```

```
SELECT
    DISTINCT profession
FROM student;
```

SELECT WHERE

```
SELECT *
FROM student
WHERE name = 'Андрій' AND surname = 'Ковальчук';
```

```
SELECT *
FROM student
WHERE profession = 121;
```

SELECT WHERE

```
SELECT *
FROM student
WHERE LOWER(name) = 'андрій';
```

```
SELECT *
FROM student
WHERE profession = 121;
```

SELECT WHERE

```
SELECT *
FROM student
WHERE LOWER(name) = 'андрій';
```

```
SELECT *
FROM student
WHERE (name = 'Андрій' OR name = 'Антон') AND profession = 121;
```

SELECT WHERE

```
SELECT *
FROM student
WHERE profession IN (121, 122);
```

```
SELECT *
FROM student
WHERE profession BETWEEN 121 AND 122;
```

SELECT WHERE IS NULL

```
SELECT *
FROM student
WHERE profession IS NULL;
```

```
SELECT *
FROM student
WHERE profession IS NOT NULL;
```

SELECT WHERE LIKE

```
SELECT *
FROM student
WHERE surname LIKE '%енко';
```

```
SELECT *
FROM student
WHERE name LIKE '_ндрій';
```

SELECT ORDER BY

```
SELECT *
FROM student
ORDER BY profession DESC NULLS FIRST;
```

```
SELECT *
FROM student
ORDER BY surname DESC, name DESC;
```

SELECT AGGREGATES

Найбільш поширені функції агрегації

- **COUNT()**
- **SUM()**
- **AVG()**
- **MIN()**
- **MAX()**

SELECT AGGREGATES

```
SELECT COUNT(*)
FROM student
WHERE profession = 121;
```

```
SELECT COUNT(profession)
FROM student;
```

```
SELECT COUNT(DISTINCT profession)
FROM student;
```

UPDATE

UPDATE <table name>
SET <column> = <value>, ...
WHERE <condition>;

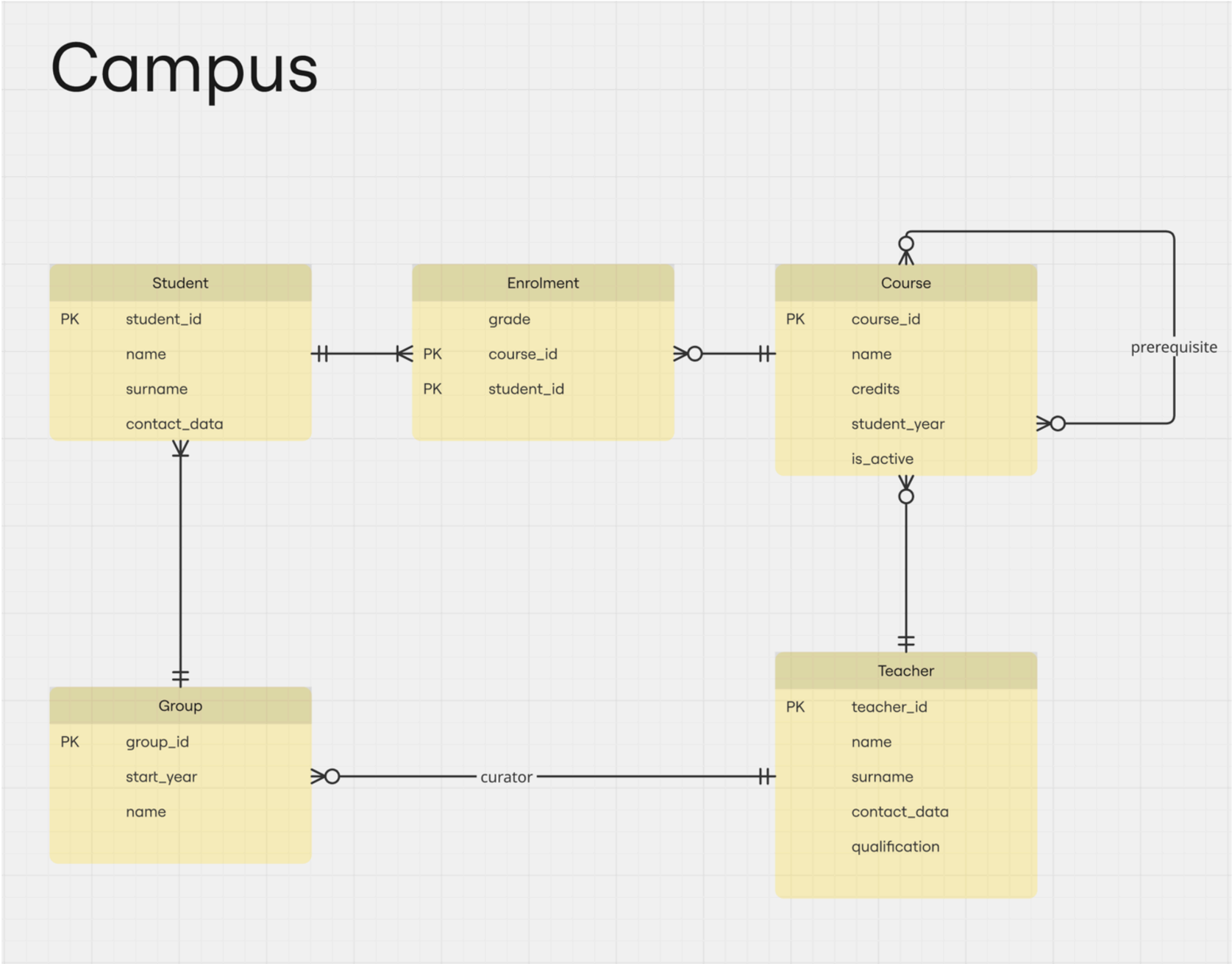
UPDATE student
SET profession = 122
WHERE student_id = 9;

DELETE

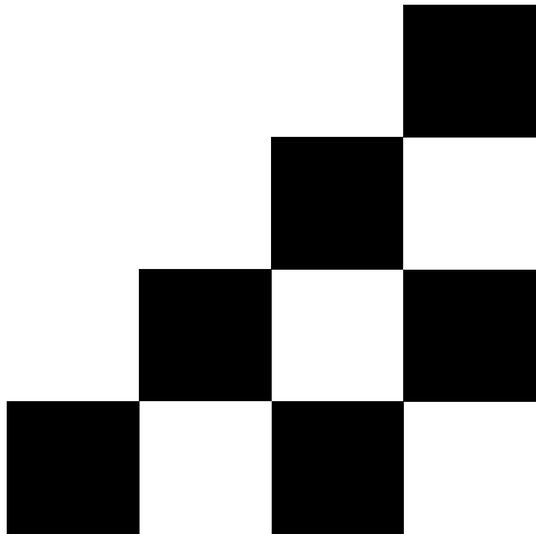
DELETE FROM <table name>
WHERE <condition>;

DELETE FROM student
WHERE student_id = 9;

Campus


Student
PK student_id
name
surname
contact_data
Enrolment
grade
PK course_id
PK student_id
Course
PK course_id
name
credits
student_year
is_active
prerequisite
Group
PK group_id
start_year
name
curator
Teacher
PK teacher_id
name
surname
contact_data
qualification

Практика
SQL запитів



Запитання