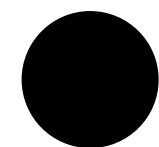


Бази даних

Лекція 7

Тематика лекції



SQL - частина 4

Підзапити (subqueries)

Common Table Expression (CTE)

Підзапити

Підзапит (вкладений запит / subquery / nested query) - запит, який є частиною іншого запиту.

Підзапити

- Підзапит виконується в першу чергу, а зовнішній запит просто використовує результат.
- Підзапит завжди повинен бути в дужках.
- Підзапити можуть бути частиною SELECT, FROM, WHERE або HAVING частин основного запиту.

Підзапити

```
SELECT student_id, course_id, grade
FROM enrolment
WHERE grade > (
    SELECT AVG(grade) FROM enrolment
);
```

Коли підзапити корисні

- Потрібно використати динамічну умову для фільтрації даних (наприклад студенти, які мають бали більші за загальний середній бал).
- Фільтрації на основі агрегованих даних.
- Перевірки існування певного запису.
- Іноді підзапити просто зручніші та простіші, ніж складні JOINи.

Види підзапитів

- Скалярні - повертають лише одне значення (1 рядок і 1 колонку).
- Колонкові - повертають багато рядків, але лише 1 колонку.
- Рядкові - повертають один рядок, але багато колонок.
- Табличні - повертають набір даних, що містить багато рядків і багато колонок.

Скалярні підзапити

```
SELECT student_id, course_id, grade
FROM enrolment
WHERE grade > (
    SELECT AVG(grade) FROM enrolment
);
```

```
SELECT
student_id, course_id, grade, (SELECT AVG(grade) FROM enrolment) as avg_grade
FROM enrolment;
```


Колонкові підзапити

```
SELECT student_id, course_id, grade  
FROM enrolment  
WHERE course_id IN (  
    SELECT course_id FROM course WHERE student_year = 2  
);
```

Колонкові підзапити

Колонкові підзапити можуть використовувати оператори:

- IN
- ANY
- ALL

Табличні підзапити

```
SELECT s.name, s.surname, avg_student_grade.avg_grade
FROM (
    SELECT student_id, AVG(grade) as avg_grade
    FROM enrolment
    GROUP BY student_id
) AS avg_student_grade
INNER JOIN student s USING (student_id)
WHERE avg_student_grade.avg_grade > 90;
```

Корельовані підзапити

- Некорельовані підзапити - незалежні від зовнішнього запиту, виконуються лише один раз.
- Корельовані підзапити - використовують колонки з зовнішніх запитів, виконуються окремо для кожного з рядків з зовнішнього запиту.

Корельовані підзапити

```
SELECT *  
FROM enrolment e  
WHERE grade > (  
    SELECT AVG(grade)  
    FROM enrolment e2  
    WHERE e.course_id = e2.course_id  
);
```

EXISTS

Оператор EXISTS повертає TRUE якщо підзапит повертає хоч один рядок.

```
SELECT *  
FROM course c  
WHERE EXISTS (  
    SELECT 1  
    FROM enrolment e  
    WHERE e.course_id = c.course_id  
);
```

Common Table Expressions (CTE)

CTE - іменовані тимчасові набори даних, що існують протягом виконання запиту.

- **CTE дозволяють перевикористовувати підзапит кілька разів в межах одного запиту.**
- **CTE допомагають краще структурувати складні запити.**
- **CTE дозволяють використовувати рекурсивні запити, що допомагає у роботі з ієрархічними даними.**

CTE

```
WITH avg_student_grade AS (  
    SELECT student_id, AVG(grade) as avg_grade  
    FROM enrolment  
    GROUP BY student_id  
)  
SELECT s.name, s.surname, asg.avg_grade  
FROM avg_student_grade asg  
INNER JOIN student s USING (student_id)  
WHERE asg.avg_grade > 90;
```

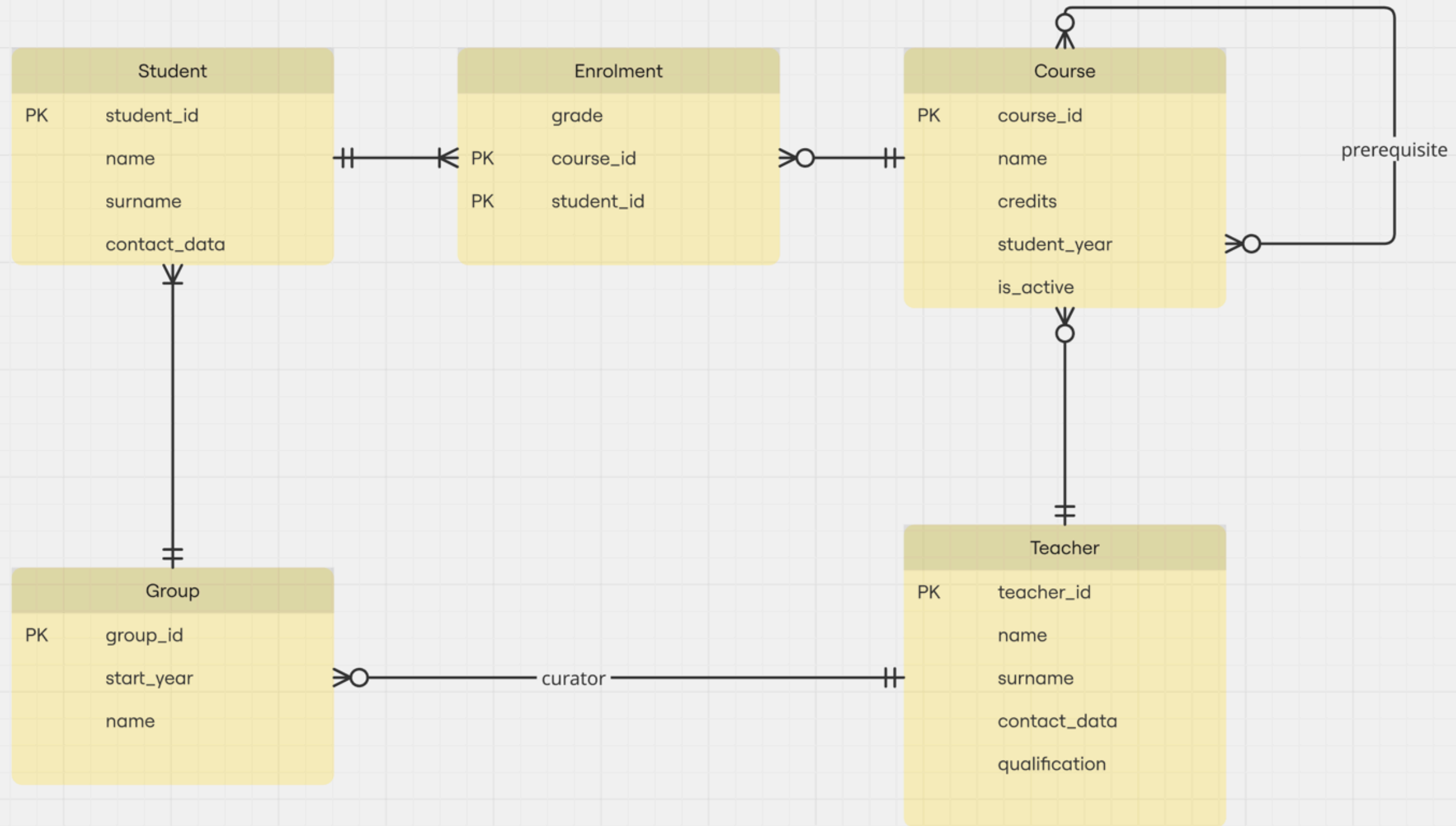

Рекурсивні СТЕ

```
WITH RECURSIVE course_dependencies AS (  
    SELECT course_id, 1 as level  
    FROM course c  
    WHERE NOT EXISTS (SELECT 1 FROM course_prerequisite p WHERE p.course_id = c.course_id)  
    UNION ALL  
    SELECT p.course_id as course_id, cd.level + 1 as level  
    FROM course_prerequisite p  
    INNER JOIN course_dependencies cd ON p.prerequisite_course_id = cd.course_id  
)  
SELECT c.name, cd.level  
FROM course c INNER JOIN course_dependencies cd USING (course_id);
```

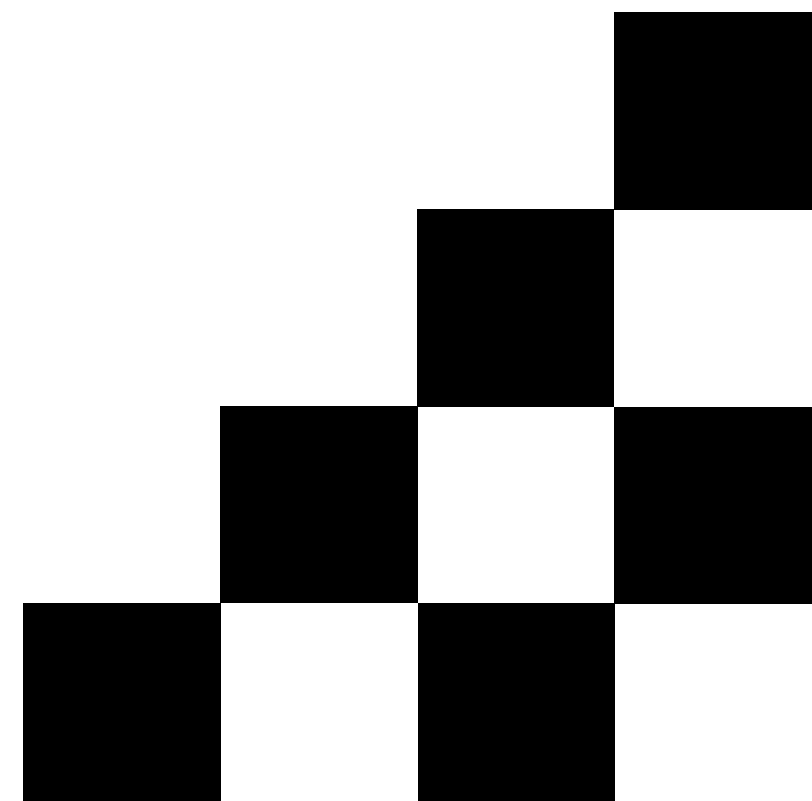
Рекомендації

- Завжди надавайте перевагу простішому та зрозумілішому запиту.
- Використовуйте CTE, якщо запит має кілька проміжних кроків.
- Використовуйте CTE, якщо запит використовує одні і ті ж обчислення кілька разів.
- Не використовуйте корельовані підзапити (окрім EXISTS).
- Використовуйте рекурсивні CTE для обробки ієрархічних даних.
- Використовуйте EXISTS + підзапит для перевірки існування.

Campus



ПРАКТИКА SQL ЗАПИТІВ



Запитання