

Л5. Проектування бази даних

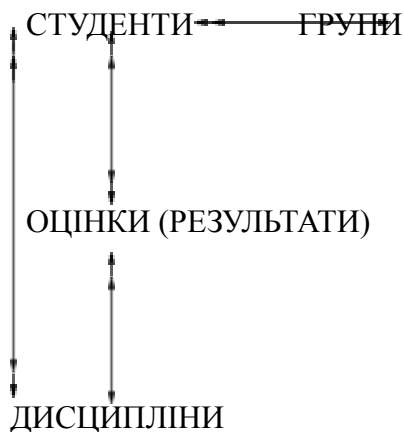
- ПОБУДОВА МОДЕЛІ

Етап 1. Визначення сутностей

Виходячи з постановки задачі, сформуємо сутності

1. **СТУДЕНТИ;**
2. **ДИСЦИПЛІНИ;**
3. **ГРУПИ;**
4. **ОЦІНКИ (РЕЗУЛЬТАТИ);**

Етап 2. Визначення взаємозв'язку між сутностями



Етап 3. Задання первинного та альтернативних ключів, визначення атрибутів сутностей

В результаті виконання третього етапу проектування отримуємо інформаційну модель БД.

ОЦІНКИ (РЕЗУЛЬТАТИ)

№зап	Студент	Дисципліна	Оцінка
1 I.C.	112	1	3
...	
8 B.B.	51	3	3

ДИСЦИПЛІНИ

Дисципліна	Назва
1	Диф.рівн.
2	Програмування
3	Мат.аналіз

СТУДЕНТИ

№студента	ПІБ	Група
112	Петренко I.C.	20
...
51	Семенов B.B.	23

ГРУПИ

№групи	Група	Староста
20	К-20	112
...

23	K-23	51
----	------	----

Мал. 5.1. Інформаційна модель БД

Етап 4. Наведення моделі до потрібного рівня нормальної форми

Реструктуризація таблиць відношень для наведення до 1/2/3 НФ. Реалізація зв'язків між відношеннями: Головна таблиця у парі відношень має первинний ключ, підлегла має атрибут, який є зовнішнім ключем (індекс за полем зв'язку).

Етап 5. Фізичний опис моделі.

Задаються імена атрибутів, визначаються типи та домени. В результаті виконання етапу маємо Фізичну модель БД.

Окремим питанням є **забезпечення цілісності** БД. Під забезпеченням цілісності бази даних розуміється система заходів, направлених на підтримку правильності даних в базі в довільний момент часу. В СУБД цілісність даних забезпечується набором спеціальних умов, які називаються обмеженнями.

Обмеження цілісності в більшості випадків визначаються особливостями предметної області. Наприклад, оцінка в національній шкалі не може бути більше 5 і менше, ніж 2.

Обмеження цілісності можуть відноситися до різних об'єктів БД: атрибутам (полям), записам (кортежам), відношенням, зв'язкам між ними і т.і. Для полів даних можуть бути застосовано такі види обмежень:

- Тип і формат поля автоматично допускають ввід лише даних конкретного типу.
- Доречним є задання значення поля за замовчанням (але не є обов'язковим).
- Для числових та символічних полів (№ телефону) логічно визначити шаблон вводу
- Вибір типу поля Date у форматі ДД.ММ.ГГ дозволяє користувачу вводити лише шість чисел. При цьому перша пара цифр не може перевищити значення 31, а друга - 12.
- Задання діапазону значень, як правило, застосовується для числових полів. Діапазон допустимих значень (домен) може бути обмежений з двох боків (закритий діапазон), а може з однієї: верхньої або нижньої (відкритий діапазон).
- Недопустимість порожнього поля дозволяє уникнути появи в БД "нічийних" записів, в яких пропущено деякі обов'язкові атрибути.
- Задання списку значень дозволяє уникнути надлишкового різноманіття даних, якщо це можна обмежити. Наприклад, для вказування значення оцінки (у довільній шкалі) можна обмежити користувача лише загально прийнятими балами: (2,3,4,5) і т.і.
- Перевірка на унікальність значення якогось поля дозволяє уникнути записів - дублікатів. Навряд чи розумно у переліку групи студентів мати декілька записів для однієї особи.

Обмеження на рівні полів застосовуються у тих випадках, коли потрібно перевіряти інформацію, яка вводиться у поле незалежно від іншої частини запису. Наприклад, можна використовувати правило перевірки значення поля, щоб виключити можливість вводу від'ємного числа у поле, яке повинно мати додатне значення. Можна застосувати правило перевірки значення поля та правило, що полягає у порівнянні введеного у поле значення зі значенням з іншої таблиці.

Не потрібно задавати правила перевірки полів і записів, якщо ці поля та записи є специфічними для даного додатку (даного варіанту використання БД). Наприклад, не має сенсу розглядати номери студентів з деякого діапазону, можна задати правило, яке буде порівнювати значення, що вводиться в поле `key_student` таблиці Оцінки, зі значеннями зі

словника студентів (СТУДЕНТ), і відкидати ті значення, які не будуть знайдені серед ключів відповідного відношення).

Забезпечення цілісності на рівні записів, таблиць або зв'язків між ними полягає у:

- забороні зміни полів зв'язку або видалення запису головної таблиці, якщо є записи для неї у підлеглий таблиці;
- забезпеченні каскадного видалення.

Такі обмеження відносяться до цілісності посилань. Для реалізації обмежень цілісності в СУБД використовуються тригери та збережені процедури.

Тригери – елементи обслуговування БД, які визначаються і зберігаються як властивості для заданої таблиці. Якщо таблиця видаляється з бази даних, то зв'язані з нею тригери також видаляються. Тригери спрацьовують після перевірки усіх інших обмежень, таких як правила перевірки значення, унікальність первинного ключа, допустимість порожніх значень.

Для таблиці може бути задано по одному тригеру на кожному події: Вставку, Зміну або Видалення. В кожен момент часу таблиця може мати максимум три тригери (відповідно до подій). Значенням тригера може бути істина (.T.) або хибність (.F.).

Для встановлення тригера задається вираз тригера або ім'я збереженої процедури, яка містить вираз тригера.

Припустимо, що при підведенні ітогів зимової сесії 27 грудня і 27 травня потрібно формувати запит про результати оцінювання в межах сесії. Щоб цього добитися, можна задати тригер на операцію Вставка для таблиці Оцінки. Тригер встановлюється на операцію Вставка, тому що необхідно, щоб тригер спрацьовував при додаванні нового запису. При створенні тригера можна вказати функцію `_insert_session()` як тригер вставки запису до таблиці Оцінки. Цю функцію необхідно описати в збереженій процедурі

```
PROCEDURE _insert_session
IF DAY(DATE())=27
WAIT WINDOW 'Формування звіту' NOWAIT
SELECT * ;
FROM Marks ;
WHERE MONTH(Marks.date_session1) = MONTH(DATE()) OR MONTH(Marks.date_session2) =
MONTH(DATE());
INTO CURSOR Session_Month_Current
ENDIF
ENDPROC
```

Інший приклад

```
CREATE PROCEDURE curdemo()
BEGIN
  DECLARE done INT DEFAULT 0;
  DECLARE a CHAR(16);
  DECLARE b,c INT;
  DECLARE cur1 CURSOR FOR SELECT id,data FROM test.t1;
  DECLARE cur2 CURSOR FOR SELECT i FROM test.t2;

  OPEN cur1;
  OPEN cur2;

  REPEAT
    FETCH cur1 INTO a, b;
    FETCH cur2 INTO c;
    IF NOT done THEN
      IF b < c THEN
        INSERT INTO test.t3 VALUES (a,b);
      ELSE
        INSERT INTO test.t3 VALUES (a,c);
      END IF;
    END IF;
  UNTIL done END REPEAT;

  CLOSE cur1;
```

```
CLOSE cur2;  
END
```

- АДМІНІСТРУВАННЯ БАЗИ ДАНИХ

Адміністратором бази даних (АБД) називається особа, відповідальна за виконання функції адміністрування бази даних.

Адміністрування бази даних передбачає обслуговування користувачів бази даних. АБД захищає ресурси, якими є дані.