

Лабораторна робота №4-5

Тема: Багатотабличні запити. Вкладені запити. Представлення.

Теоретичні відомості:

Об'єднання таблиць однакової структури

Оператор **UNION** використовується для об'єднання записів таблиць, які мають однакові структури, виключаючи дублікати. Для виведення усіх даних разом з **UNION** використовується оператор **ALL**.

Приклад:

Вивести прізвища, імена і по батькові усіх студентів і викладачів.

```
SELECT SFAM, SIMA, SOTCH FROM STUDENTS  
UNION [ALL]  
SELECT VFAM, VIMA, VOTCH FROM VYKLAD;
```

Встановлення зв'язків між таблицями

При використанні декількох таблиць у запиті, їх потрібно об'єднати по ключових полях.

Перший спосіб полягає у заданні рівності ключових полів в умові **WHERE**:

Приклад:

Вивести прізвища студентів і їх оцінки.

```
SELECT STUDENTS.SFAM, USPISH.OCINKA  
FROM STUDENTS, USPISH  
WHERE STUDENTS.SNOM=USPISH.SNOM;
```

Другий спосіб полягає у встановленні зв'язків між таблицями по ключових полях:

INNER JOIN – встановлює об'єднання, в якому вибираються тільки ті записи, які містять співпадаючі значення в полях зв'язку;

LEFT [OUTER] JOIN – встановлює лівостороннє зовнішнє об'єднання, тобто таке, в якому вибираються всі записи з лівої таблиці і записи з правої таблиці, для яких значення поля зв'язку співпадає із значенням поля зв'язку лівої таблиці;

RIGHT [OUTER] JOIN – встановлює правостороннє зовнішнє об'єднання, тобто таке, в якому вибираються всі записи з правої таблиці і записи з лівої таблиці, для яких значення поля зв'язку співпадає із значенням поля зв'язку правої таблиці;

FULL [OUTER] JOIN – створює повне зовнішнє об'єднання, в якому вибираються всі значення із лівої і правої таблиць.

Приклад:

Вивести прізвища студентів і їх оцінки.

```
SELECT STUDENTS.SFAM, USPISH.OCINKA  
FROM STUDENTS INNER JOIN USPISH  
ON STUDENTS.SNOM=USPISH.SNOM;
```

Використання трьох таблиць в запиті

Приклад:

Вивести прізвища викладачів, дату іспиту і назву предмета

```
SELECT DISTINCT VYKLAD.VFAM, USPISH.DATA,
```

```
PREDMET.PNAME
FROM VYKLAD INNER JOIN PREDMET INNER JOIN USPISH
ON USPISH.PNOM=PREDMET.PNOM
ON PREDMET.VNOM=VYKLAD.VNOM;
```

Використання вкладених запитів

Вкладені запити створюються шляхом розміщення одного запиту (підзапиту) в середині другого. Підзапит повертає завжди одне значення.

Приклад:

Вивести номери дисциплін і оцінки, вищі від середньої.

```
SELECT PNOM, OCINKA
FROM USPISH
WHERE OCINKA > (SELECT AVG(OCINKA) FROM USPISH);
```

У вкладених запитах можуть використовуватись оператори **EXIST**, **ANY**, **ALL**, **SOME**, які в якості аргументу використовують підзапит.

Результатом оператора **EXIST** є значення „істинно”, якщо він здійснює будь-яке виведення записів, і „хибно” в протилежному випадку. Підзапит, який використовується в якості аргументу оператора **EXIST**, може повертати декілька значень або жодного.

Приклад:

Вивести дані з таблиці USPISH, якщо в ній є відмінні оцінки.

```
SELECT * FROM USPISH WHERE USPISH.OCINKA=5 AND EXIST
(SELECT * FROM USPISH WHERE USPISH.OCINKA=5);
```

Існує ще три оператори, що орієнтовані на підзапити – це **ANY**, **ALL** та **SOME**. Але вони відрізняються від **EXIST** тим, що використовуються разом з реляційними операторами в підзапитах. Оператори **ANY** та **SOME** взаємозамінні, тому в поданих прикладах будуть працювати однаково. Оператор **ALL** вибирає всі значення, що виведенні підзапитом.

Приклад:

Вивести інформацію про студентів, що отримали оцінки з різних навчальних предметів.

```
SELECT * FROM STUDENTS WHERE SNOM = ANY (SELECT SNOM
FROM USPISH);
```

Даний запит можна виконати за допомогою оператора **IN**.

```
SELECT * FROM STUDENTS WHERE SNOM IN (SELECT SNOM
FROM USPISH);
```

Приклад:

Вивести дані про тих студентів, чії оцінки вище або рівні отриманим 13.06.2005

```
SELECT * FROM USPISH WHERE OCINKA >= ALL (SELECT OCINKA
FROM USPISH WHERE DATA='13.06.2005')
```

Представлення

Представлення – це віртуальна таблиця, створена на основі запиту до реальних таблиць. Представлення виконується кожен раз, коли відбувається до нього звертання.

1. Створення представлення

```
CREATE VIEW ім'я_представлення [(ім'я_поля [, ім'я_поля ...])]  
AS <select> [WITH CHECK OPTION];
```

ім'я_поля – задає ім'я поля в представленні. Якщо дане ім'я відсутнє, то вибирається по замовчуванню ім'я стовпця з основної таблиці. Кількість імен стовпців, визначені в представленні, має відповідати кількості і порядку стовпців, перерахованих в <select>.

<select> - команда select, яка задає критерії вибору записів.

WITH CHECK OPTION – забороняє модифікацію представлень (застосування команд INSERT, DELETE або UPDATE відповідно до представлень), якщо порушуються умови, задані в конструкції WHERE.

За допомогою модифікації представлень можна змінювати вміст основних таблиць. Але це можливо в тому випадку, якщо усі поля таблиці допускають наявність NULL-значень.

Представлення не можуть включати конструкцію **ORDER BY**.

Приклад:

Створити представлення, яке містить інформацію про студентів, які отримують стипендію.

```
CREATE VIEW STIPEN (NAME, STIPENDIA) AS  
SELECT SFAM, STIP  
FROM STUDENTS  
WHERE STIP<>0;
```

2. Знищення представлення

```
DROP VIEW ім'я_представлення;
```

Представлення можна видалити в тому випадку, якщо воно не використовується в інших представленнях.

Завдання до виконання:

- 1) Завантажте програму *IBExpert*.
- 2) Відкрийте базу даних **Univer**.
- 3) Використовуючи команду **SELECT** у командному вікні *SQL Editor*, виконайте наступні запити:
 1. Вивести прізвища, імена, по батькові студентів і викладачів, відсотувати їх в алфавітному порядку по прізвищу.
 2. До запиту 1 додати вивід стовпця *POSADA*, який включає значення 'студент' і 'викладач' відповідно для кожного запису
 3. Вивести прізвища викладачів і назви предметів, які вони викладають.
 4. Вивести прізвища викладачів, назви предметів, які вони викладають і кількість годин по кожному предмету, якщо кількість годин >100.
 5. За допомогою команди **INSERT** доповнити таблицю **STUDENTS** записами про двох нових студентів. Вивести прізвища усіх студентів і номери предметів, з яких вони здали іспити. (лівостороннє об'єднання).
 6. Записати попередній запит з використанням правостороннього об'єднання.
 7. Вивести прізвища студентів і назви предметів, іспити з яких вони повинні скласти, тобто розклад іспитів (операція декартового добутку).

8. Вивести прізвища студентів і назви предметів, іспити з яких вони склали. (задати зв'язок між таблицями).
9. Використовуючи вкладені запити, вивести інформацію про предмети, які викладає викладач Бачишина.
10. Використовуючи вкладені запити, вивести прізвища студентів, які отримують максимальну стипендію (розмір максимальної стипендії наперед невідомий).
11. Використовуючи оператори EXIST, ANY, ALL, SOME, сформувані запити, що реалізують наступні завдання:
 - 1) Вивести дані про студентів, що мають незадовільні оцінки
 - 2) Вивести прізвища тих студентів, що мають лише одну „2” .
 - 3) Вивести дані про студентів, що здали всі іспити.
 - 4) Вивести прізвища тих викладачів, що читають більше ніж один предмет.
 - 5) Вивести прізвища всіх студентів, що здавали іспит у вказаний день.
 - 6) Вивести назви тих дисциплін, для вивчення яких відведена однакова кількість годин.
 - 7) Вивести прізвища тих студентів, що мають ім'я, яке співпадає із заданим.
 - 8) Вивести прізвища всіх студентів, які здавали іспити (без повторень).

Контрольні запитання:

- 1) Яке призначення оператора UNION?
- 2) Яким чином встановлюється зв'язок між двома таблицями в запитах?
- 3) Які є види зв'язків між таблицями?
- 4) Що таке вкладені запити?
- 5) Які оператори використовуються в вкладених запитах?
- 6) Що таке представлення?
- 7) Чим представлення відрізняються від запитів?
- 8) Чи може бути створене представлення на основі іншого представлення?
- 9) Чи зберігаються дані, отримані в представленнях, у базі даних?