**7 Мова структурованих запитів SQL**

**Лабораторна робота №7**

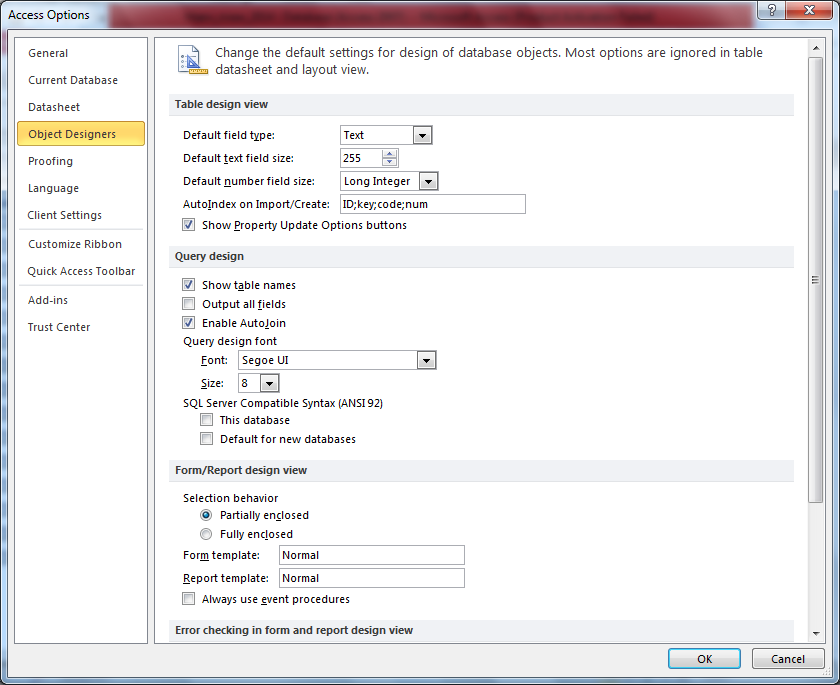
Мова SQL є найбільш поширеною мовою визначення (DDL) та маніпулювання (DML) в сучасних СКБД.

Теоретичний матеріал з описом форматів команд мови SQL викладений в матодичних матеріалах випуск №122 „Створення та ведення баз даних засобами мови Jet SQL. Методичні вказівки до виконання блоку лабораторних робіт з дисципліни „Організація баз даних” (автори М.Т.Фісун, С.Г.Ніколенко).

СКБД Access має свою реалізацію SQL – Jet SQL.

В лабораторній роботі наведени приклади створення запитів цим діалектом мови SQL. В СКБД Access вона реалізована як інтерпретатор, тобто виконується кожний оператор мови, ознакою закінчення якого є символ крапка з комою “;”.

СКБД Access 2010 може створювати запити Jet SQL в форматі SQL1 (по замовченню) та в форматі SQL2. Режим SQL2 вказується в команді OPTIONS – Object Designers вкладки File:



Лабораторна робота використовує такі таблиці:

**DYSCYPL1 –** перелік дисциплін;

**VYKLADACH1 –** список викладачів;

**POSADA1 –** перелік посад викладачів;

**ZAKR\_DYSC –** закріплення викладачів за дисциплінами.

У наведених таблицях будуть використовуватися такі атрибути:

**KodDysc –** код дисципліни;

**NazvDysc –** назва дисципліни;

**KodVykl –** код викладача;

**PrizvVykl –** прізвище код викладача;

**NomKaf –** номер кафедри;

**KodPost –** код посади код викладача;

**PostVykl –** назва посади код викладача;

**NormPost –** норма навантаження для викладача даної посади.

**7.1 Керуючі оператори (команди) Jet SQL (Access)**

Нижче перелічені керуючі оператори Jet SQL – службові слова початку оператора та його суть:

CREATE TABLE – створити таблицю (визначити структуру таблиці);

CREATE INDEX – створити індекс, в т.ч. складений;

DROP TABLE – видалити таблицю(і);

DROP INDEX – видалити індекс;

ALTER TABLE – змінити структури таблиці (додати поле –ADD, видалити – DROP);

INSERT – вставити (додати) дані (рядок або рядки із іншої таблиці/запиту);

UPDATE – оновити дані (поля) в таблиці;

DELETE – видалити рядки із таблиці;

**7.1.1 CREATE TABLE** – створити таблицю

***Приклади***

7.1.1.1. CREATE TABLE **DYSCYPL1** (**KodDysc** TEXT (5) PRIMARY KEY, **NazvDysc** TEXT (50) NOT NULL);

У режимі конструктора структуру цієї таблиці показано на рис. 7.1.

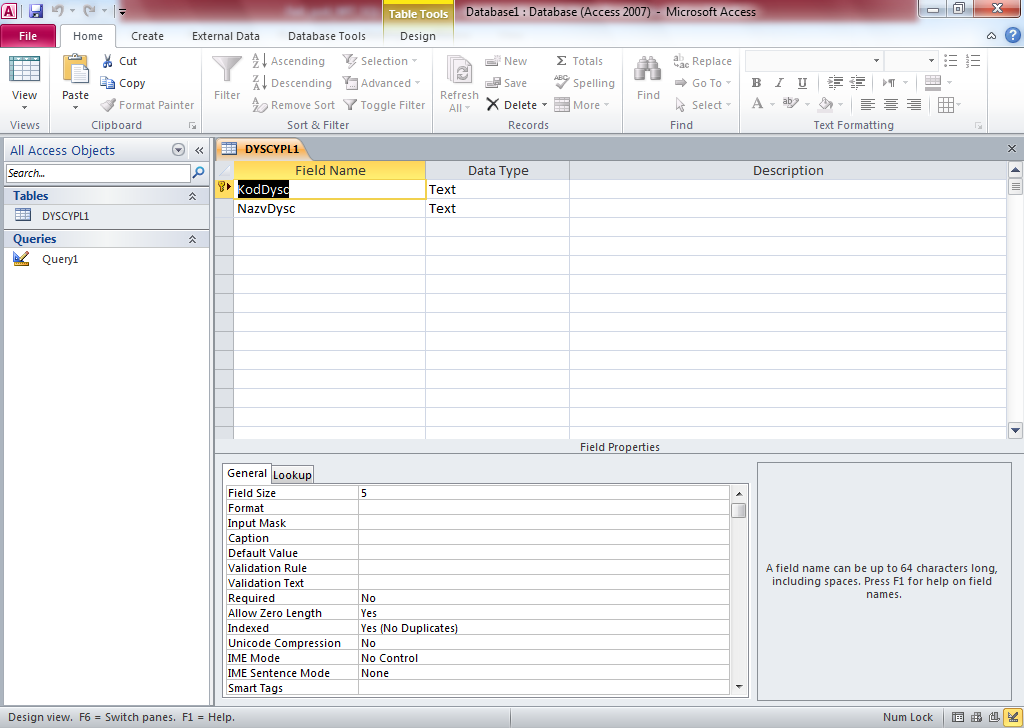


Рис. 7.1. Структура таблиці **DYSCYPL1**

7.1.1.2. CREATE TABLE **VYKLADACH1** (**KodVykl** SMALLINT PRIMARY KEY, **PrizvVykl** TEXT (25), **KodPost** TEXT (2), **NomKaf**  BYTE);

У режимі конструктора структуру цієї таблиці показано на рис. 7.2.

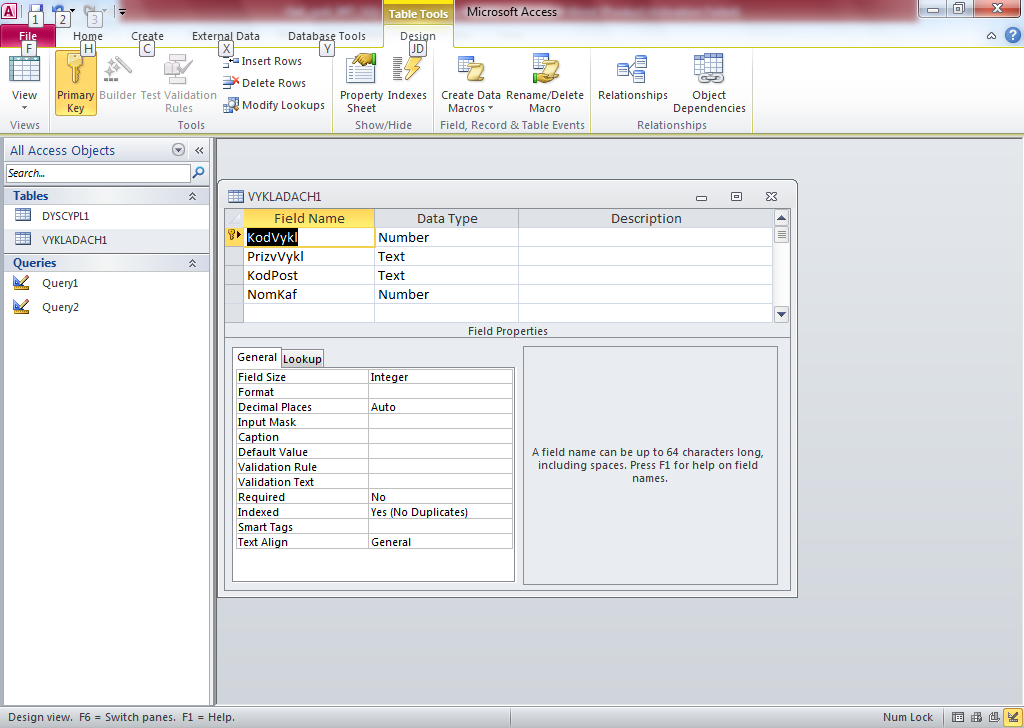


Рис. 7.2. Структура таблиці **VYKLADACH1**

7.1.1.3**.** CREATE TABLE **ZAKR\_DYSC** (**KodVykl** INTEGER, **PrizvVykl** TEXT (25), **PostVykl** TEXT (25), **KodDysc** TEXT (5), **NazvDysc** TEXT (50), CONSTRAINT **Zakr** PRIMARY KEY (**KodVykl**, **KodDysc**) );

У режимі конструктора структуру цієї таблиці показано на рис. 7.3.

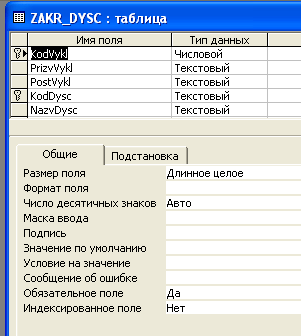


Рис. 7.3. Структура таблиці **ZAKR\_DYSC**

7.1.1.4. CREATE TABLE **POSADA**1 (**KodPost** TEXT (2) PRIMARY KEY, **PostVykl** TEXT (25), **NormPost** NUMBER NOT NULL);

У режимі конструктора структуру цієї таблиці показано на рис. 7.3.

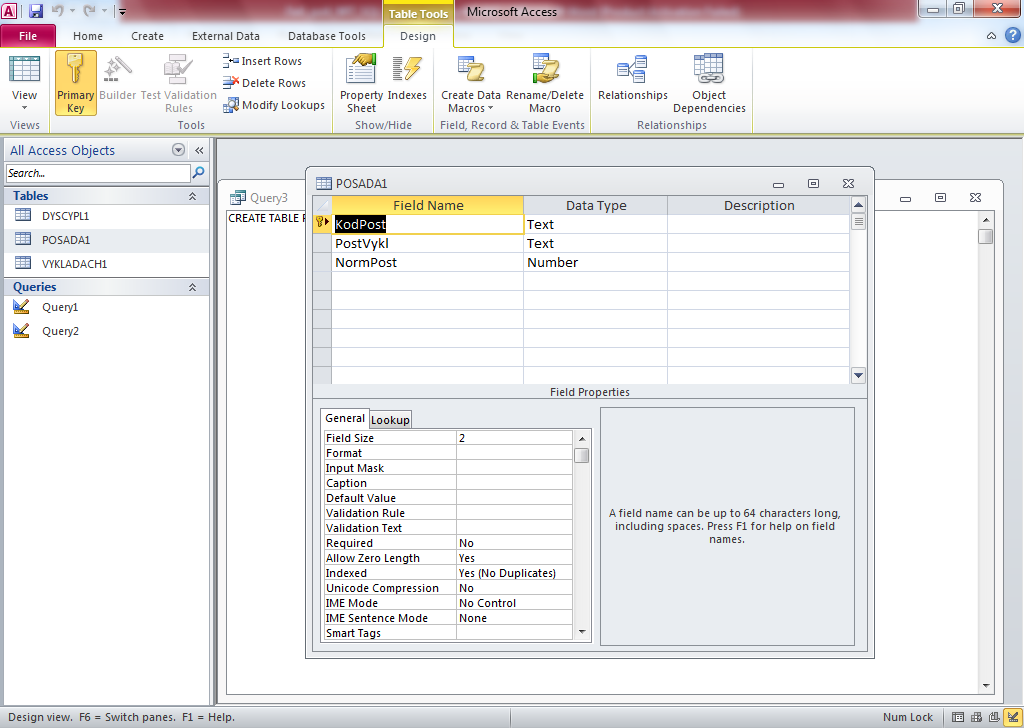


Рис. 7.4. Структура таблиці **POSADA1**

**7.1.2 CREATE INDEX** – створити індекс

**Приклади:**

7.1.2.1 CREATE UNIQUE INDEX **PostVykl** ON **POSADA1** (**PostVyk**l);

Результат показано на рис.7.5.

Не унікальний індекс можна створити командою CREATE INDEX.

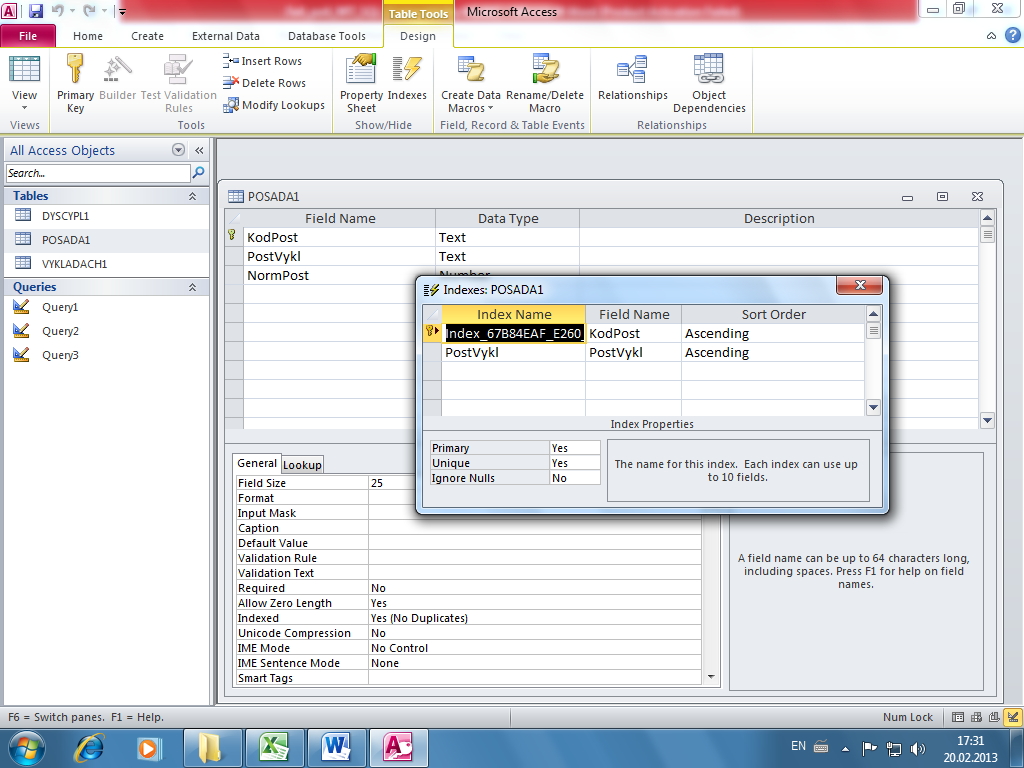


Рис. 7.5. Результат індексації поля **PostVykl** таблиці **POSADA1**

7. 1.2.2 CREATE INDEX **NmKf** ON **VYKLADACH1** (**NomKaf**);

Результат показано на рис.7.6. Щоб отримати більш детальну інформацію, крім тієї, що видно із структури таблиці (ім’я індексу та його склад), треба при відкритій у режимі конструктора таблиці зайти на панель інструментів **Вид→Индекси.**

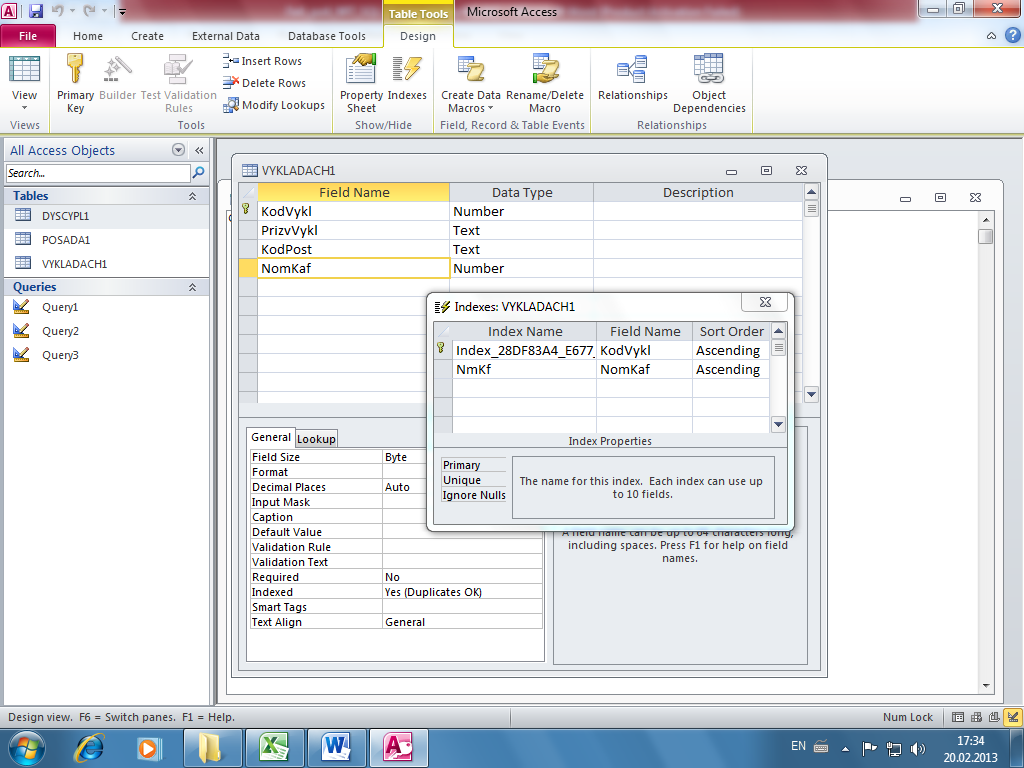


Рис.7.6 Результат індексації поля **NomKaf** таблиці **VYKLADACH1**

7. 1.2.3 CREATE INDEX **KafDysc** ON **ZAKR\_DYSC**(**KodVykl, KodDysc**);

Результат індексації показано на рис.7.7.

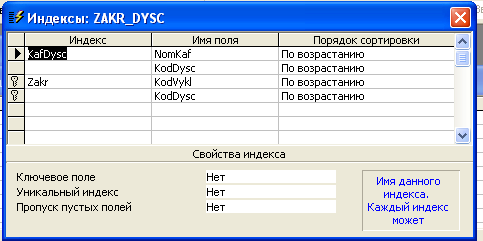


Рис.7.7 Результат індексації полів (**NomKaf, KodDysc**) таблиці **ZAKR\_DYSC**

Як бачимо, таблиця **ZAKR\_DYSC** вже має два індекси. Індекс **Zakr** , який складається із полів (**KodVykl**, **KodDysc**), був створений автоматично при визначенні цій сукупності полів як PRIMARY KEY при створенні таблиці (п. 7.1.1.3).

**7.1.3 DROP TABLE** - видалити таблицю

**Приклад:**

DROP TABLE **DYSCYPL1;**

**7.1.4 DROP INDEX** - видалити індекс

**Приклад:**

DROP INDEX **KafDysc** ON  **ZAKR\_DYSC;**

Інформація про індекси таблиці наведено на рис.7.8. Як бачимо, в ній залишився один індекс **Zakr** , який був створений автоматично при встановленні первинного ключа.

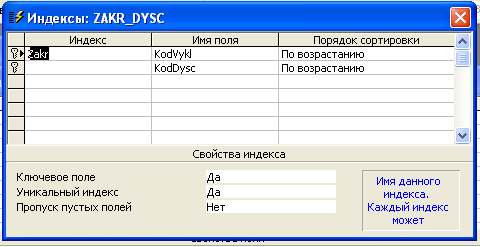


Рис.7.8 Індекси таблиці **ZAKR\_DYSC** після видалення індексу **KafDysc**

**7.1.5 ALTER TABLE** - змінити структури таблиці

**Приклади**:

7.1.5.1. Додати 2 поля і складений індекс до таблиці **ZAKR\_DYSC**

ALTER TABLE **ZAKR\_DYSC** ADD  **KodPost** TEXT (2), **NomKaf** BYTE, **NazvKaf** TEXT (25),CONSTRAINT Idx1 UNIQUE (**KodVykl**, Nazv**Dysc**) );

У режимі конструктора нова структура таблиці показано на рис. 7.9, а її індекси – на рис. 7.10.

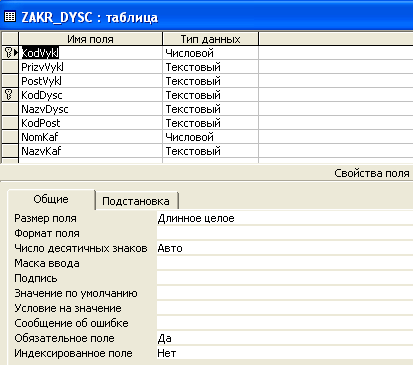


Рис.7.9. Структура таблиці **ZAKR\_DYSC**

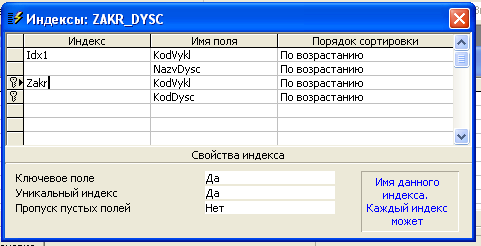


Рис. 7.10. Первинний ключ і складені індекси таблиці **ZAKR\_DYSC**

7.1.5.2. Видалимо індекс **Idx1** із таблиці **ZAKR\_DYSC.**

ALTER TABLE **ZAKR\_DYSC** DROP CONSTRAINT **Idx1**;

Після виконання цього запиту структура індексів повернеться до стану, що приведений на рис.7.8.

7.1.5.3. Видалимо поле **NazvKaf** із таблиці **ZAKR\_DYSC.**

ALTER TABLE **ZAKR\_DYSC** DROP **NazvKaf**;

Примітка. Запити 7.1.5.2 і 7.1.5.3 можна було об’єднати в один запит:

ALTER TABLE **ZAKR\_DYSC** DROP CONSTRAINT Idx1, COLUMN **NazvKaf**;

7.1.5.4 Надамо полю **NormPost** таблиці **POSADA1** властивість необов’язкового значення (NULL). Для цього прийдеться спочатку вилучити його, а потім додати з новою властивістю, без якої буде неможливим виконання операції п.1.6.2.

*1-й запит*. ALTER TABLE **POSADA1** DROP COLUMN **NormPost**;

*2-й запит*. ALTER TABLE **POSADA**1 ADD COLUMN NormPost NUMBER NULL;

**7.1.6 INSERT** –додати рядок або рядки із іншої таблиці/запиту

**Приклади:**

7.1.6.1 Додати до таблиці **POSADA1** рядок

|  |  |  |
| --- | --- | --- |
| 01 | професор | 700 |

INSERT INTO **POSADA**1 VALUES ("**01**","**професор**",**700**);

7.1.6.2 Додати до таблиці **POSADA1** рядок

|  |  |
| --- | --- |
| 02 | доцент |

INSERT INTO **POSADA1** (**KodPost,** **PostVykl)** VALUES ("**02**","**доцент**");

Після виконання операторів п.1.6.1 і 1.6.2 таблиця **POSADA1** матиме такий вміст

|  |  |  |
| --- | --- | --- |
| **KodPost** | **PostVykl** | **NormPost** |
| 01 | професор | 700 |
| 02 | доцент |  |

|  |  |  |
| --- | --- | --- |
| 01 | професор | 700 |
| 02 | доцент |  |
| 03 | старший викладач | 850 |
| 04 | викладач | 900 |

* + - 1. Приклад команди INSERT, що має формат INSERT …SELECT. В таблицю треба вставити дані з іншої таблиці:

INSERT INTO DSCPL ( CdDs1, TtDs2, DsCrr, CdClnn, CdCf )

SELECT DISC.CdDs, DISC.TtDs, DISC.DsCr, DISC.CdCl, DISC.CdCf

FROM DISC

WHERE (DISC.CdDs="17") OR (DISC.CdDs="33") OR (DISC.CdDs="34");

Або вставити в таблицю **VYKLADACH1** всі записи з таблиці **VYKLADACH** з відповідними полями

INSERT INTO VYKLADACH1 ( KodVykl, PrizvVykl, KodPost, NomKaf )

SELECT VYKLADACH.KodVykl, VYKLADACH.PrizvVykl, VYKLADACH.KodPost, VYKLADACH.NomKaf

FROM VYKLADACH;

* + 1. **UPDATE** - оновити дані (поля) в таблиці

**Приклади:**

7.1.7.1 Увести дані про норму навантаження доцента (код ”02”)

UPDATE **POSADA1** SET **NormPost** = **650** WHERE **KodPost** =”02”;

7.1.7.2 Зменшити норму для усіх посад, крім доцента, на 100 годин

UPDATE **POSADA1** SET **NormPost** = **NormPost-100** WHERE **KodPost** <>”02”;

Після виконання наведених вище операторів вміст таблиці буде таким.

|  |  |  |
| --- | --- | --- |
| **KodPost** | **PostVykl** | **NormPost** |
| 01 | професор | 600 |
| 02 | доцент | 650 |
| 03 | старший викладач | 750 |
| 04 | викладач | 800 |

Умова виконання запиту UPDATE може бути побудована з використанням полів інших таблиці, але для цього треба задіяти операції з’єднання (сполучення) таблиць, про які мова піде в п.2 (запити SELECT).

* + 1. **DELETE** – видалити рядки із таблиці

**Приклад:**

З таблиці VYKLADACH1 вилучимо кафедри №5.

DELETE **FROM VYKLADACH1** WHERE **NomKaf=5;**

**Завдання. Частина 1.**

* + 1. Ознайомитися з прикладами керуючих операторів і виконати КР №4.

**Завдання. Частина 2.**

* 1. **Запити мовою Jet SQL на селекцію (вибірку та об’єднання) даних, на створення розрахункових полів, на групування даних.**

# 7.2.1.Отримати список викладачів та їх посад :

SELECT VYKLADACH.PrizvVykl, POSADA.PostVykl, POSADA.NormPost

FROM VYKLADACH INNER JOIN POSADA ON VYKLADACH.KodPost=POSADA.KodPost;

# 7.2.2.Теж саме для кафедри №7 так:

SELECT VYKLADACH.PrizvVykl, POSADA.PostVykl, POSADA.NormPost, VYKLADACH.NomKaf

FROM POSADA INNER JOIN VYKLADACH ON POSADA.KodPost = VYKLADACH.KodPost

WHERE (((VYKLADACH.NomKaf)=7));

| **Query3** | | | |
| --- | --- | --- | --- |
| **Прізвище викладача** | **Назва посади** | **Норма** | **Код кафедри** |
| Колісніченко А.І. | професор, доктор наук | 410 | 7 |
| Яжборовська | старший викладач без наукового ступеню | 630 | 7 |
| Ярошенко | старший викладач без наукового ступеню | 630 | 7 |

7.2.3. Зробимо запит до таблиць VYKLADACH і POSADA, при цьому зменшимо норму навантаження викладачів на 10%. Ім’я розрахункового поля будемо задавати за допомогою конструкції “**AS** <ім’я атрибута>”.

SELECT VYKLADACH.KodVykl, VYKLADACH.PrizvVykl, POSADA.PostVykl, 0.9\*[NormPost] AS Нова\_норма

FROM VYKLADACH INNER JOIN POSADA ON VYKLADACH.KodPost = POSADA.KodPost;

| **Query4** | | | |
| --- | --- | --- | --- |
| **Код викладача** | **Прізвище викладача** | **Назва посади** | **Нова\_норма** |
| 0101 | Тригуб | професор, доктор наук | 369 |
| 0202 | Дихта | професор, доктор наук | 369 |
| 0204 | Сердюченко | професор, доктор наук | 369 |
| 0501 | Фісун | професор, доктор наук | 369 |
| 0511 | Кондратенко | професор, доктор наук | 369 |
| 0514 | Мєщанінов | професор, доктор наук | 369 |
| 0701 | Колісніченко А.І. | професор, доктор наук | 369 |
| 0801 | Грабак | професор, доктор наук | 369 |
| 1201 | Казарєзов | професор, доктор наук | 369 |
| 1401 | Дубова | професор, доктор наук | 369 |
| 1501 | Горлачук | професор, доктор наук | 369 |
| … |  |  |  |

# 7.2.4.Підрахувати кількість дисциплін циклу „3” у 5-му триместрі.

## SELECT DYSCYPL.KodCycl, DYS\_TRYM.NomTrym, Count(DYS\_TRYM.KodDysc) AS [кількість дисциплін]

## FROM DYSCYPL INNER JOIN DYS\_TRYM ON DYSCYPL.KodDysc = DYS\_TRYM.KodDysc

## GROUP BY DYSCYPL.KodCycl, DYS\_TRYM.NomTrym

## HAVING (((DYSCYPL.KodCycl)=3) AND ((DYS\_TRYM.NomTrym)=5));

## 

# 7.2.5.Яка кафедра має найбільшу кількість дисциплін у навчальному плані на 2 курсі?

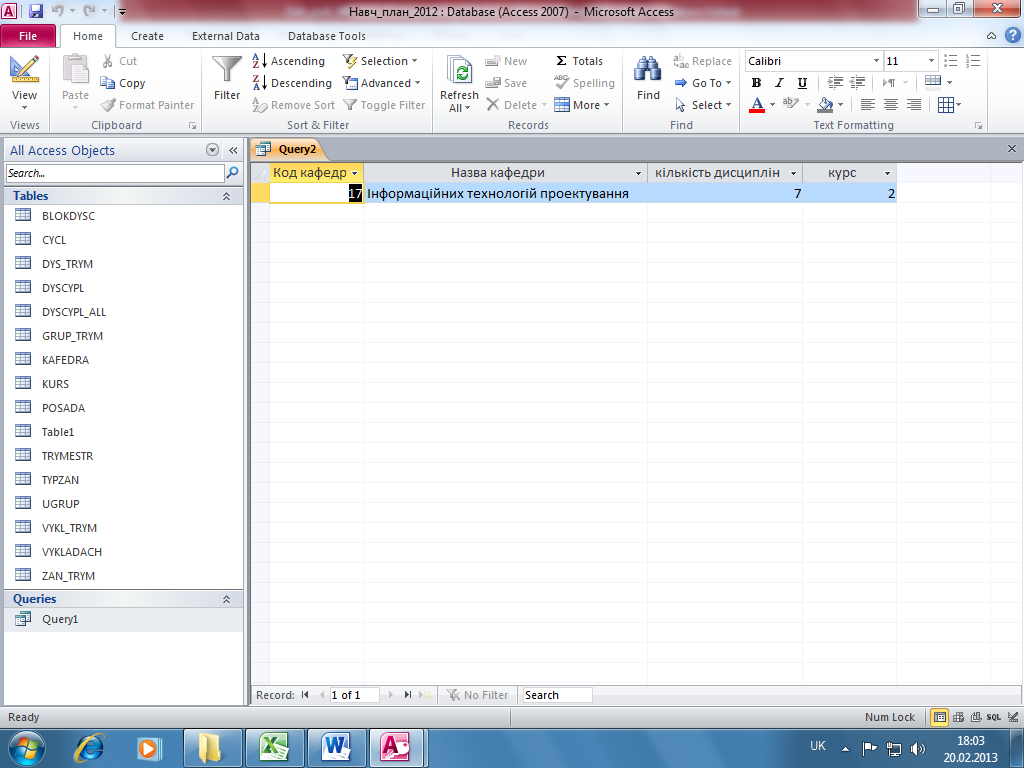
# SELECT TOP 1 DYSCYPL.NomKaf, KAFEDRA.NazvKaf, Count(DYS\_TRYM.KodDysc) AS [кількість дисциплін], TRYMESTR.NomKurs

# FROM KAFEDRA INNER JOIN (TRYMESTR INNER JOIN (DYSCYPL INNER JOIN DYS\_TRYM ON DYSCYPL.KodDysc = DYS\_TRYM.KodDysc) ON TRYMESTR.NomTrym = DYS\_TRYM.NomTrym) ON KAFEDRA.NomKaf = DYSCYPL.NomKaf

# GROUP BY DYSCYPL.NomKaf, KAFEDRA.NazvKaf, TRYMESTR.NomKurs

# HAVING (((TRYMESTR.NomKurs)=2))

# ORDER BY Count(DYS\_TRYM.KodDysc) DESC;



* 1. **Структуровані запити та підзапити**

**7.3.1 Підзапити з оператором порівняння**

Розглянемо, наприклад, задачу складання списку всіх викладачів, що мають мінімальну норму навантаження (ми не знаємо коду чи назви посади, що має мінімальну норму). За допомогою з’єднань цю задачу можна було б виконати в два прийоми.

1. Знайти мінімальну норму навантаження.

SELECT MIN(NormPost)

FROM POSADA;

| **Min-NormPost** |
| --- |
| 600 |

2. Одержати прізвища викладачів, які мають таку норму.

SELECT VYKLADACH.PrizvVykl, POSADA.PostVykl, POSADA.NormPost

FROM VYKLADACH, POSADA

WHERE price = 600;

Якщо зробити це з допомогою підзапиту, то нам буде потрібно тільки один оператор:

SELECT VYKLADACH.PrizvVykl, POSADA.PostVykl, POSADA.NormPost

FROM VYKLADACH INNER JOIN POSADA ON VYKLADACH.KodPost = POSADA.KodPost

WHERE (((POSADA.NormPost)=(select MIN(NormPost) from POSADA)));

**7.3.2 Підзапити з предикатом IN (Not In)**

Запит – знайти усіх викладачів, які мають усі ті посади, що є на кафедрі №2 .

SELECT VYKLADACH.PrizvVykl, VYKLADACH.NomKaf, VYKLADACH.KodPost

FROM VYKLADACH

WHERE ((( VYKLADACH.KodPost) In (select KodPost from VYKLADACH where NomKaf=2)));

| **PrizvVykl** | **NomKaf** | **KodPost** |
| --- | --- | --- |
| Бандур М.М. | 6 | 02 |
| Сірко В.В. | 7 | 02 |
| Опришко Ю.Й. | 2 | 02 |
| Батура О.Й. | 6 | 02 |
| Балагура А.М. | 2 | 04 |
| Черно Ф.М. | 2 | 04 |

А якщо треба, щоб в результатній таблиці не було викладачів кафедри №2, то

SELECT VYKLADACH.PrizvVykl, VYKLADACH.NomKaf, VYKLADACH.KodPost

FROM VYKLADACH

WHERE ((( VYKLADACH.NomKaf)<>2) AND (( VYKLADACH.KodPost) In (select KodPost from VYKLADACH where NomKaf=2)));

| **PrizvVykl** | **NomKaf** | **KodPost** |
| --- | --- | --- |
| Бандур М.М. | 6 | 02 |
| Сірко В.В. | 7 | 02 |
| Батура О.Й. | 6 | 02 |

*Запит* – знайти усіх викладачів, які мають посади, яких немає на кафедрі №2 .

SELECT VYKLADACH.PrizvVykl, VYKLADACH.NomKaf, VYKLADACH.KodPost

FROM VYKLADACH

WHERE ((( VYKLADACH.KodPost) Not In (select KodPost from VYKLADACH where NomKaf=2)));

| **PrizvVykl** | **NomKaf** | **KodPost** |
| --- | --- | --- |
| Середа І.С. | 6 | 01 |
| Мунтян Р.Д. | 7 | 01 |
| Турчак А.Д. | 12 | 03 |
| Махно Н.Є. | 7 | 01 |
| Собчак А.О. | 12 | 01 |

7.3.5 Підзапити, що починаються з операторів порівняння та ключових слів ANY або ALL

*7.5.5.1 Підзапити з ключовим словом ANY*

*Запит* : знайти посади викладачів, що більше (за кодом, а не за семантичним значенням) будь якої посади викладачів кафедри №12 (на кафедрі дві посади з кодами „01” і „03”).

SELECT VYKLADACH.PrizvVykl, VYKLADACH.NomKaf, VYKLADACH.KodPost

FROM VYKLADACH

WHERE ((( VYKLADACH.KodPost)>Any (select KodPost from VYKLADACH where NomKaf=12)));

| **PrizvVykl** | **NomKaf** | **KodPost** |
| --- | --- | --- |
| Бандур М.М. | 6 | 02 |
| Сірко В.В. | 7 | 02 |
| Опришко Ю.Й. | 2 | 02 |
| Турчак А.Д. | 12 | 03 |
| Батура О.Й. | 6 | 02 |
| Балагура А.М. | 2 | 04 |
| Черно Ф.М. | 2 | 04 |

Як бачимо, в результат попали усі рядки, де **KodPost** >”01”. Змінимо оператор порівняння на “<”.

| **PrizvVykl** | **NomKaf** | **KodPost** |
| --- | --- | --- |
| Середа І.С. | 6 | 01 |
| Бандур М.М. | 6 | 02 |
| Мунтян Р.Д. | 7 | 01 |
| Сірко В.В. | 7 | 02 |
| Опришко Ю.Й. | 2 | 02 |
| Батура О.Й. | 6 | 02 |
| Махно Н.Є. | 7 | 01 |
| Собчак А.О. | 12 | 01 |

Як бачимо, в результат попали усі рядки, де **KodPost** <”04”. Змінимо оператор порівняння на “=”.

SELECT VYKLADACH.PrizvVykl, VYKLADACH.NomKaf, VYKLADACH.KodPost

FROM VYKLADACH

WHERE ((( VYKLADACH.KodPost)=Any (select KodPost from VYKLADACH where NomKaf=12)));

| **PrizvVykl** | **NomKaf** | **KodPost** |
| --- | --- | --- |
| Середа І.С. | 6 | 01 |
| Мунтян Р.Д. | 7 | 01 |
| Турчак А.Д. | 12 | 03 |
| Махно Н.Є. | 7 | 01 |
| Собчак А.О. | 12 | 01 |

Результат цього запиту повністю співпав би із результатом запиту з предикатом **In**. Покажемо це на запиті для кафедри №2 (див вище запит для цієї кафедри з предикатом **In)**.

SELECT VYKLADACH.PrizvVykl, VYKLADACH.NomKaf, VYKLADACH.KodPost

FROM VYKLADACH

WHERE ((( VYKLADACH.KodPost) =Any (select KodPost from VYKLADACH where NomKaf=2)));

| **PrizvVykl** | **NomKaf** | **KodPost** |
| --- | --- | --- |
| Бандур М.М. | 6 | 02 |
| Сірко В.В. | 7 | 02 |
| Опришко Ю.Й. | 2 | 02 |
| Батура О.Й. | 6 | 02 |
| Балагура А.М. | 2 | 04 |
| Черно Ф.М. | 2 | 04 |

*Приклади.*

SELECT VYKLADACH.PrizvVykl, VYKLADACH.NomKaf, VYKLADACH.KodPost

FROM VYKLADACH

WHERE ((( VYKLADACH.KodPost) <>Any (select KodPost from VYKLADACH where NomKaf=2)));

| **PrizvVykl** | **NomKaf** | **KodPost** |
| --- | --- | --- |
| Середа І.С. | 6 | 01 |
| Бандур М.М. | 6 | 02 |
| Мунтян Р.Д. | 7 | 01 |
| Сірко В.В. | 7 | 02 |
| Опришко Ю.Й. | 2 | 02 |
| Турчак А.Д. | 12 | 03 |
| Батура О.Й. | 6 | 02 |
| Балагура А.М. | 2 | 04 |
| Махно Н.Є. | 7 | 01 |
| Собчак А.О. | 12 | 01 |
| Черно Ф.М. | 2 | 04 |

*7.5.5.2 Підзапити з ключовим словом ALL*

*Запит* : знайти посади викладачів, що більше (за кодом, а не за семантичним значенням) самої більшої (за кодом) посади викладачів кафедри №12 (на кафедрі дві посади з кодами „01” і „03”).

SELECT VYKLADACH.PrizvVykl, VYKLADACH.NomKaf, VYKLADACH.KodPost

FROM VYKLADACH

WHERE ((( VYKLADACH.KodPost)>All (select KodPost from VYKLADACH where NomKaf=12)));

| **PrizvVykl** | **NomKaf** | **KodPost** |
| --- | --- | --- |
| Балагура А.М. | 2 | 04 |
| Черно Ф.М. | 2 | 04 |

Як що оператор порівняння змінимо на “<All”, то отримаємо пусту таблицю, оскільки посади з кодом, меншим за „01” в таблиці VYKLADACH не існує. Якщо ми зробимо аналогічний запит для кафедри №2 (там є посади з кодами “02” і “04”), то отримаємо список викладачів код посади у яких має значення “01”.

SELECT VYKLADACH.PrizvVykl, VYKLADACH.NomKaf, VYKLADACH.KodPost

FROM VYKLADACH

WHERE ((( VYKLADACH.KodPost) <All (select KodPost from VYKLADACH where NomKaf=2)));

| **PrizvVykl** | **NomKaf** | **KodPost** |
| --- | --- | --- |
| Середа І.С. | 6 | 01 |
| Мунтян Р.Д. | 7 | 01 |
| Махно Н.Є. | 7 | 01 |
| Собчак А.О. | 12 | 01 |

При вживанні операторів порівняння “=All” і “<>All” отримаємо пусті таблиці, оскільки для таблиць ці оператори не визначені. Крім того, якщо внутрішній підзапит, що починається з ALL і оператора порівняння, повертає в якості одного зі своїх значень NULL, вважається, що запит у цілому завершився невдало.

*Приклад.* Для демонстрації вищенаведеного змінимо значення поля KodPost для викладача з номером 22 на Null (він працює на кафедрі №12, яка буде фігурувати у внутрішньому запиті), а також занесемо одне значення Null для викладача с номером 19 (він працює на кафедрі №2, яка буде оброблятися зовнішнім запитом).

UPDATE VYKLADACH SET VYKLADACH.KodPost = Null

WHERE ((( VYKLADACH.KodVykl)=22) or (( VYKLADACH.KodVykl)=19));

Тепер на кафедрі №12, на якій працює два викладача, код посади залишився тільки у одного і він дорівнює „03”, у другого – не визначений, а на кафедрі №2 із трьох викладачів у двох коди посад “02” “04”, у третього - – не визначений.

| **VYKLADACH** | | | |
| --- | --- | --- | --- |
| **KodVykl** | **PrizvVykl** | **KodPost** | **NomKaf** |
| 24 | Черно Ф.М. | 04 | 2 |
| 19 | Балагура А.М. |  | 2 |
| 15 | Опришко Ю.Й. | 02 | 2 |
| 18 | Батура О.Й. | 02 | 6 |
| 5 | Бандур М.М. | 02 | 6 |
| 3 | Середа І.С. | 01 | 6 |
| 21 | Махно Н.Є. | 01 | 7 |
| 11 | Сірко В.В. | 02 | 7 |
| 6 | Мунтян Р.Д. | 01 | 7 |
| 22 | Собчак А.О. |  | 12 |
| 17 | Турчак А.Д. | 03 | 12 |

Повторимо запит з використанням предикатів “>All” та “<All” (див. вище) для оновленої таблиці VYKLADACH.

SELECT VYKLADACH.PrizvVykl, VYKLADACH.NomKaf, VYKLADACH.KodPost

FROM VYKLADACH

WHERE ((( VYKLADACH.KodPost) >All (select KodPost from VYKLADACH where NomKaf=12)));

| **PrizvVykl** | **NomKaf** | **KodPost** |
| --- | --- | --- |
|  |  |  |

Результат – пуста таблиця, хоча в таблиці VYKLADACH є посади з кодом „04”, який більше ніж „03”, але старшинство між „Null” і „04”, як і будь-яким іншим значенням коду, не визначено системою. Такий же результат отримаємо, коли замінимо предикат на “<All”, хоча в таблиці VYKLADACH є посади з кодом „01” і „02”, який менше ніж „03”, але не визначено їх порівняння з „Null”.

Повернемося до попереднього стану таблиці VYKLADACH.

UPDATE VYKLADACH SET VYKLADACH.KodPost = “01”

WHERE (( VYKLADACH.KodVykl)=22);

UPDATE VYKLADACH SET VYKLADACH.KodPost = “04”

WHERE (( VYKLADACH.KodVykl)=19));

**7.5.6 Підзапити, які починаються з EXISTS або NOT EXISTS**

Як вже доводилося вище, ці підзапити представляють собою тест на існування рядків чи не існування рядків в результаті виконання внутрішнього запиту. Якщо відповідь позитивна, то виконується зовнішній запит, у протилежному випадку він не виконується, тобто результатна таблиця буде пустою. Таким чином, застосування квантора EXISTS аналогічно оператору розгалуження в процедурних мовах програмування IF <*умова*> THEN <*дія*>, але складові конструкції стоять у зворотному порядку: <*дія*> IF <*умова*>. Звідси витікає, що після ключового слова WHERE відсутній атрибут для порівняння, після WHERE одразу пишеться ключове слово EXISTS, після якого в дужках записується оператор SELECT. Тому зрозуміло правило, за яким після опції SELECT можна ставити символ “\*”, тому що немає значення, скільки атрибутів ми задамо ­– результатна таблиця буде пустою або не пустою, що і визначає, чи буде включатися зовнішній оператор SELECT, чи ні.

*Приклад.* Отримати список викладачів кафедри №2 та їх посад при умові, що на цій кафедрі є хоча б один викладач, код посади якого „02” (доцент).

SELECT VYKLADACH.PrizvVykl, POSADA.PostVykl

FROM VYKLADACH INNER JOIN POSADA ON VYKLADACH.KodPost = POSADA.KodPost

WHERE ((( VYKLADACH.NomKaf)=2) AND ((Exists (SELECT \* FROM VYKLADACH where ((KodPost="02") AND (NomKaf=2))))=True));

| **PrizvVykl** | **PostVykl** |
| --- | --- |
| Опришко Ю.Й. | доцент |
| Балагура А.М. | викладач |
| Черно Ф.М. | викладач |

### Ті ж самі результати отримаємо, якщо при кванторі Exists в останньому рядку запиту поміняємо умову “=True” на “<>False”, або замість квантора Exists застосуємо квантор Not Exists разом з умовою “<>True” або “=False”.

Якщо в наведеному вище запиті змінимо тільки Exists на Not Exists, то отримаємо пусту таблицю.

### SELECT VYKLADACH.PrizvVykl, POSADA.PostVykl

### FROM VYKLADACH INNER JOIN POSADA ON VYKLADACH.KodPost = POSADA.KodPost

### WHERE ((( VYKLADACH.NomKaf)=2) AND ((Not Exists (SELECT \* FROM VYKLADACH where ((KodPost="02") AND (NomKaf=2))))=True));

| **PrizvVykl** | **PostVykl** |
| --- | --- |
|  |  |

Як і в попередньому запиті, існує ще три комбінації із { Not Exists, Exists, True, False, =, <>}, при яких будемо отримувати пусту результатну таблицю.

**7.6 Лабораторна робота №7. Частина 3. Завдання**

1. Визначити перелік дисциплін 5-го триместру, які не мають таких видів занять, як дисципліна „Організація баз даних” (код 3.06) (тобто, мають інші види занять крім лекцій лабораторних та інше ).
2. Визначити перелік дисциплін 6-го триместру, які мають тижневий обсяг лекційних занять більшим або рівним обсягу лекційних занять будь-якої дисципліни 5-го триместру.
3. Визначити перелік дисциплін кафедри №17, які мають обсяг занять більшим або рівним обсягу дисципліни кафедри №14 з максимальним обсягом.
4. Перелік дисциплін 4-го триместру при умові, якщо в ньому немає курсових робіт.
5. Кінець роботи