

#### Зміст

- ▶ Основи роботи з SQL\*plus
  - Запуск
  - Форматування виводу
  - Запис результатів у файл
  - Перегляд структури таблиці
- Прості SQL-запити
  - ▶ Вибрати всі записи з таблиці Dept
  - ▶ Вибрати частину атрибутів з таблиці Dept
- Операції з таблицями
  - Створення таблиці
  - Видалення таблиці
  - <u>Створення таблиць за ER-моделлю</u>
- ▶ 3віт
- Контрольні питання
- Домашне завдання
- 1) Познайомитися з <u>описом основних типів даних</u>

## Завдання 1. Основи роботи з SQL\*plus

Скачайте клієнтський додаток розархівуйте і запустіть SQL.bat

```
Він встановлює параметри і з'єднує з базою даних:

SET LINESIZE 200 — кількість символів у рядку екрану

SET PAGESIZE 140 — кількість рядків на сторінці

SET NULL (null) — що виводити замість порожніх значень
```

Користувачи: student\_pm81\_01 ... student\_pm81\_20 Пароль: student

Sqlplus student\_pm81\_01/student@195.182.202.156 — приєднається до бази даних на комп'ютері за вказаною IP-адресою.

## У лабораторній роботі використовуються таблиці :

**EMP** 

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
	7698	KING BLAKE	PRESIDENT MANAGER		17-NOV-81 01-MAY-81	2850		10
	7566	CLARK JONES MARTIN	MANAGER MANAGER SALESMAN	7839	09-JUN-81 02-APR-81 28-SEP-81	2975	1400	10 20 30
	7499	ALLEN TURNER	SALESMAN SALESMAN	7698	20-FEB-81 08-SEP-81	1600	300	30 30
DEPT		JAMES WARD	CLERK		03-DEC-81 22-FEB-81		500	30 30
DEI	PTNO DNAME		LOC	7902	03-DEC-81 17-DEC-80	800		20 20
	10 ACCOUN 20 RESEAR 30 SALES	СН	NEW YORK  DALLAS  CHICAGO	7788	09-DEC-82 12-JAN-83	3000 1100		20 20
	40 OPERAT		BOSTON	7782	23-JAN-82	1300		10

1. Виведемо всі записи з таблиці DEPT:

```
SELECT * FROM Dept ;
```

 Якщо виводиться повідомлення про помилку «немає таблиці» ії потрібно створити. Для цього завантажте в середовище і виконайте файл Init\_script.sql (команда @).

2. Переглянемо структуру таблиці DEPT:

```
DESCRIBE dept;
```

3. Виведемо всі назви відділів з таблиці Dept

```
SELECT dname FROM Dept;
```

4. Виведемо всі назви відділів та їх розташування з таблиці Dept:

```
SELECT dname, loc FROM Dept;
```

## Завдання 2. Зміна порядку стовпців

```
SELECT [DISTINCT] {*, column [alias],...}
FROM table;
```

5. Запит завдання 1 вивів стовпці в послідовності deptno, dname, loc. Скорегуйте запит так, щоб він виводив стовпці в порядку deptno, loc, dname.



## Завдання 2. Запит з використанням DISTINCT

```
SELECT [DISTINCT] {*, column [alias],...}
FROM table;
```

- 6. Складіть запит, який поверне всі можливі посади співробітників *без повторень*.
- 7. Складіть запит, який поверне всі можливі міста розміщення відділів *без повторень*.



# Завдання 3. Особливості SQL \* plus

# Команди редагування SQL \* Plus (не дуже зручно, використовуйте EDIT)

Команда	Дія		
A[PPEND] текст	Додати текст в кінець поточного рядка.		
C[HANGE] /старий/новий/	Замінити в поточному рядку старий текст на новий.		
C[HANGE] /meкcm/	Замінити текст в буфері SQL		
CL[EAR] BUFF[ER]	Видалити всі рядки з буферу SQL		
DEL	Видалити поточний рядок		
DEL n	Видалити рядок, що заданий параметром <i>п</i>		
DEL m n	Видалити рядки від <i>m</i> до <i>n</i>		

# Команди редагування SQL \* Plus

Команда	Опис
I[NPUT]	Вставити невизначену кількість рядків
I[NPUT] text	Вставити рядок, що складається з тексту.
L[IST]	Вивести список всіх рядків в буфері SQL.
L[IST] n	Вивести один рядок (з номером $n$ ).
L[IST] m n	Вивести діапазон рядків від n до m.
R[UN]	Вивести і виконати команду з буферу SQL.
1	Синонім RUN
n	Вказати рядок, який повинен стати поточним.
n text	Замінити рядок <i>п</i> текстом.
0 text	Вставити рядок перед рядком 1

## Файлові команди SQL \* Plus

Команда	Опис
SAV[E]  im's_фаила[.ext]  [REP[LACE] APP[END]]	Зберігає в файлі поточний зміст буфера SQL в файлі. APPEND використовується для додавання інформації в наявний файл. REPLACE замінює наявний файл. За замовчуванням файл має розширення .sql.
GET ім'я_файла.[ext]	Поміщає вміст раніше збереженого файлу в буфер SQL. За замовчуванням файл має розширення .sql.
START filename	Запускає виконання раніше збереженого файлу команд.
@ filename	Синонім START
EDIT	Викликає редактор і зберігає вміст буфера в файлі afiedt.buf.
EDIT filename	Викликає редактор для редагування збереженого файлу.
SPOOL filename	Записує результати запиту в файл. OFF закриває буферний файл (спул-файл). OUT закриває буферний файл і відправляє результати з файлу на системний принтер.
EXIT	Вихід з SQL * Plus (Синонім QUIT)

## Команда SPOOL (Зручно виконати на початку лаб. роботи, щоб швидше оформити результат)

SPO[OL] [file name[.ext] [CRE[ATE] | REP[LACE] | APP[END]] | OFF | OUT]

Аргумент	Значення
file_name[.ext]	Запис результатів у файл
CRE[ATE]	Створює новий файл із заданим ім'ям
REP[LACE]	Перезаписує файл, якщо він існує. Якщо не існує - створює і перезаписує
APP[END]	Додає виведений результат у кінець файлу
OFF	Припиняє вивід у файл
OUT	Припиняє вивід і намагається роздрукувати на принтері

#### Завдання 3. Виведення результатів в форматі html

Виведіть результати запиту в форматі html, перегляньте результат.

```
set markup html on
spool htmldemo.htm replace
select * from user_tables;
spool off
set markup html off
```

# Завдання 4. Створення таблиць

## Створення таблиць

```
CREATE TABLE STUDENT(-- коментар
STUDENT_ID INTEGER,
SURNAME VARCHAR (25), -- ще один коментар
KURS INTEGER,
BIRTHDAY /* і ще один коментар */ DATE
);
```

## Вимоги до імен таблиць та атрибутів

- Ім'я повинно починатися з символу.
- Може мати довжину 1-30 символів.
- ▶ Може містити тільки символи А–Z, а–z, 0–9, \_, \$, #.
- Не може співпадати з ім'ям другого об'екту, що належить цьому ж користувачеві.
- ▶ Не може збігатися з зарезервованими командами Oracle Server.

#### Однакові

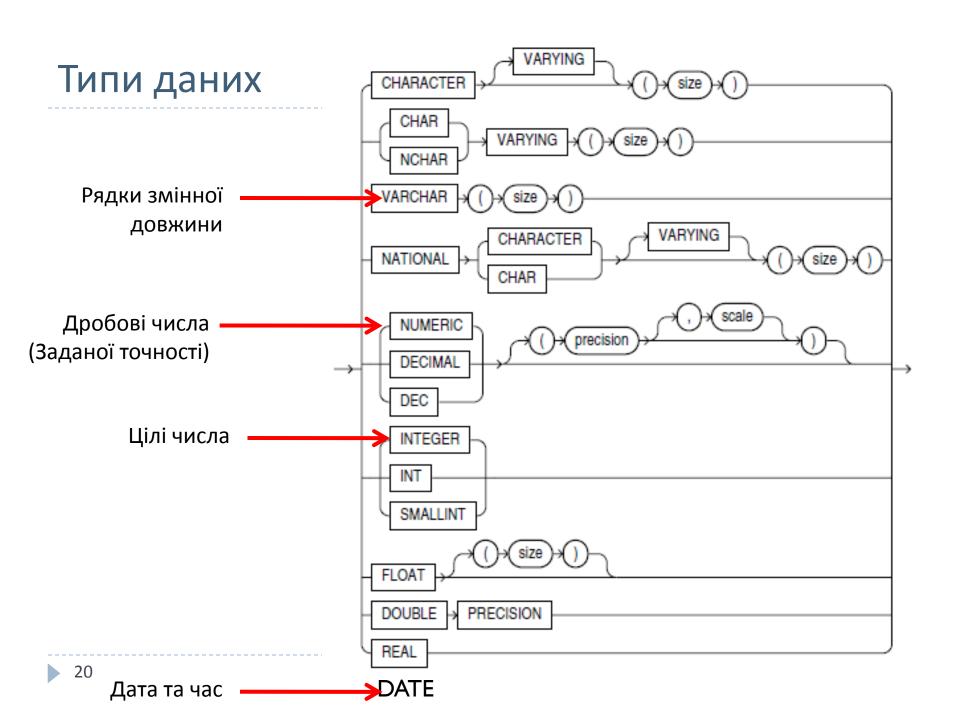
- employees
- EMPLOYEES
- "EMPLOYEES"

#### Різні

- "employees"
- "Employees"
- "EMPLOYEES"

## Типи даних

Тип	Опис
VARCHAR2(size)	Символьні дані змінної довжини
CHAR(size)	Символьні дані фіксованої довжини
NUMBER(p,s)	Чисельні дані змінної довжини
DATE	Дата та час
LONG	Символьні дані змінної довжини до 2ГБ
CLOB	Символьні дані (1 байт) данные до 4ГБ
RAW и LONG RAW	Бінарні дані
BLOB	Бінарні дані до 4ГБ
BFILE	Бінарні дані, що зберігаються в зовнішньому файлі, до 4ГБ



## Обмеження на дані

```
CREATE TABLE STUDENT (
STUDENT ID INTEGER NOT NULL UNIQUE,
SURNAME VARCHAR (25) NOT NULL,
NAME VARCHAR (10) NOT NULL,
                                      Значення поля
STIPEND INTEGER
                                      унікальне в межах
                                      таблиці
KURS INTEGER,
CITY VARCHAR (15),
                             Поле не може
                             бути порожнім
BIRTHDAY DATE,
UNIV ID INTEGER,
                                      Значення набору
UNIQUE (SURNAME, BIRTHDAY)
                                      полів унікальне в
);
                                      межах таблиці
```

## Обмеження на дані

```
CREATE TABLE STUDENT(
SURNAME VARCHAR (25),
NAME VARCHAR (10),
STIPEND INTEGER,
CITY VARCHAR (15) DEFAULT 'SUMY',
                                     Створюємо
BIRTHDAY DATE,
                                     обмеження
                                     типу «CHECK»
UNIV ID INTEGER,
CONSTRAINT STUD CHECK CHECK (STIPEND > 100500
 AND CITY = 'SUMY'
                             Надаємо
                             обмеженню ім'я
                             (не обов'язково)
```

## Завдання 4.

### Створення таблиць з обмеженнями

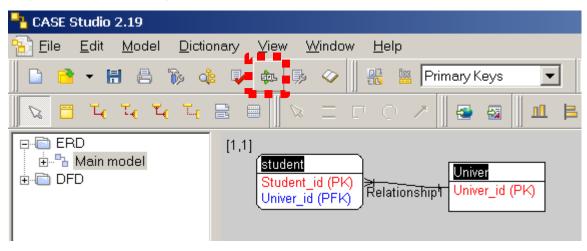
```
CREATE TABLE STUDENT (
STUDENT_ID INTEGER,
SURNAME CHAR (25),
NAME CHAR (10),
STIPEND INTEGER,
KURS INTEGER,
CITY CHAR (15),
BIRTHDAY DATE,
UNIV_ID INTEGER
);
```

- ▶ Візьміть за основу код SQL для створення таблиці STUDENT і створіть таблицю STUDENT1:
- ▶ Пара значень (Прізвище, Курс) є унікальною;
- Стипендія не може бути менше ніж 5000;
- Поле Student\_id не може бути порожнім;
- ▶ За замовчуваннямUNIV ID 0;
- ► Переглянте структуру таблиці STUDENT1;
- ▶ Видаліть таблицю STUDENT1.

# Завдання 5. Виконання сценаріїв

## Завдання 5. Створення таблиць из ERD

 1) Візьміть код створення таблиць із завдання 1-ої лабораторної роботи;



- 2) Запустіть скрипт в SQL \* plus;
- 3) Перевірте, що таблиці створені;
- 4) Видаліть створені таблиці.

#### Звіт

- Для виконання лабораторної роботи вам необхідно здійснити завдання 1-5.
- Звіт повинен містити запити та результати, необхідні для виконання завдання 2, 4.
- Звіт виконайте за шаблоном, нумерацію завдань вказуйте як на слайді.

#### Контрольні запитання

- Чим запити SQL відрізняються від команд SQL \* Plus?
- Які види обмежень на значення полів таблиці ви знаєте? Як їх створити?
- Як додати в таблицю первинний ключ? Зовнішній ключ?
- Наведіть простий запит для створення таблиці.
- Наведіть простий запит для видалення таблиці.
- Як можна переглянути структуру таблиці?
- Чи чутливі запити SQL до регістру?
- Як задати псевдонім колонки?
- Чи є різниця між псевдонімами ЕМР\_Name, "ЕМР\_NAME", "ЕМР name"?
- Чи можна розбивати SQL запити на кілька рядків?
- Чи впливають пробіли та табуляції на результат запиту?
- ▶ Що означає \* у виразі Select?