



# Бази даних та Інформаційні системи

## 4.1 Створення та управління таблицями

СумДУ, каф. КН  
2020

# Задачі заняття

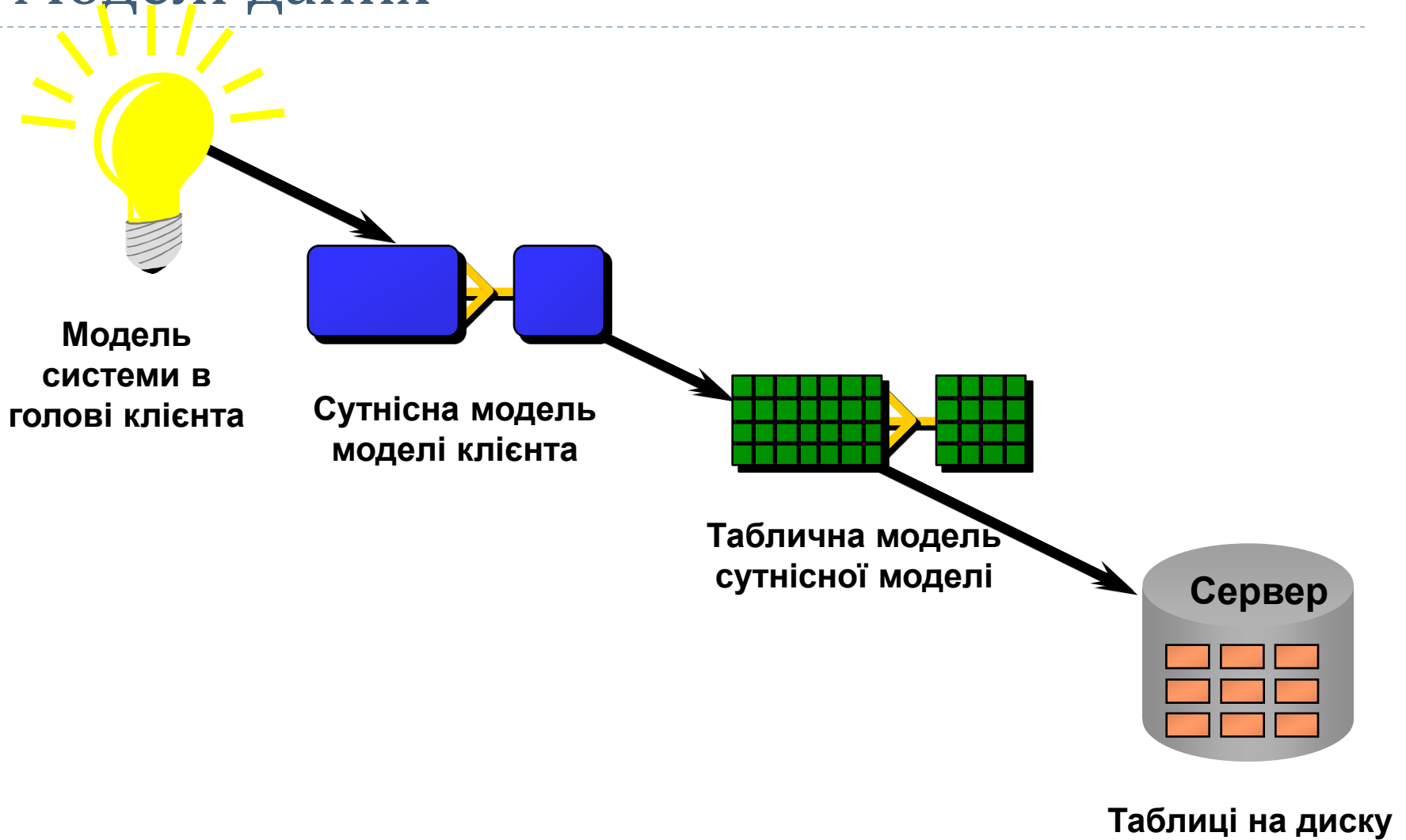
---

- ▶ Після завершення заняття ви повинні вміти і знати:
  - ▶ Основні об'єкти БД;
  - ▶ Як створювати таблиці;
  - ▶ Типи даних, які можуть бути використані при визначенні стовпця;
  - ▶ Змінювати схеми таблиць;
  - ▶ Видаляти, перейменовувати і очищати таблиці.



# Моделі даних

---



# ERD в структуру таблиц

---

1. Кожна проста сутність стає таблицею.
2. Кожен атрибут стає стовпчиком таблиці.  
Обов'язковим атрибутам встановлюють обмеження not null.
3. Ідентифікатори сутностей стають ключами. Один з ідентифікаторів стає первинним ключем.
4. Зв'язки «один-до-одного» і «один-до-багатьох» перетворюються в зовнішні ключі.



# Термінологія реляційних БД

2	3	4					
EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
-----	-----	-----	6	-----	-----	-----	-----
7839	KING	PRESIDENT		17-NOV-81	5000		10
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7900	JAMES	CLERK	7698	03-DEC-81	950		30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7902	FORD	ANALYST	7566	03-DEC-81	3000		20
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7788	SCOTT	ANALYST	7566	09-DEC-82	3000		20

# Властивості реляційних БД

---

## Реляційна БД

- Доступ до даних і їх модифікація здійснюється за допомогою мови запитів (SQL);
- Містить набір таблиць, який не залежить від фізичного розташування (користувач не знає як і де вони зберігаються).





# Взаємодія з БД за допомогою SQL

Вводиться вираз на SQL

```
SQL> SELECT loc  
2 FROM dept;
```

Вираз надсилається

в БД

БД

Виводяться дані

LOC
NEW YORK
DALLAS
CHICAGO
BOSTON

# Oracle11g: об'єктна реляційна СКБД

---

- Визначені користувачем типи даних і об'єкти
- Повна реляційна сумісність
- Підтримка мультимедійних та великих об'єктів
- Широкі можливості сервера

Базова Інструкція -

[SQL Language Reference \(e2608-8\)](#) (1680 стр)





# Мова SQL

---

- ▶ **SQL** (*Structured query language*) — структурована мова запитів.
- ▶ Складається з:
- ▶ **DDL** (Data Definition Language) — робота зі структурою бази;
- ▶ **DML** (Data Manipulation Language) — робота з даними;
- ▶ **DCI** (Data Control Language) — робота з правами;
- ▶ **TCL** (Transaction Control Language) — робота з транзакціями.



# Написання виразів SQL

---

- ▶ Вирази SQL не чутливі до регістру.
- ▶ Вирази SQL можуть складатися з одного або декількох рядків.
- ▶ Ключові слова не можуть переноситися і скорочуватися.
- ▶ Табуляції і відступи використовуються для зручності при читанні.



# Об'єкти бази даних

---

Об'єкт	Опис
Table	<b>Елемент зберігання; складається з рядків і стовпців</b>
View	Логічно представляє (подає) підмножину
Sequence	Генерує значення первинних ключів
Index	Збільшує швидкість роботи
Synonym	Альтернативне ім'я об'єкта



# Угода про надання імен

---

- ▶ Ім'я повинно починатися з символу.
- ▶ Може мати довжину 1-30 символів.
- ▶ Може містити тільки символи A-Z, a-z, 0-9, \_, \$, і #.
- ▶ Не може збігатися з ім'ям іншого об'єкта, що належить цьому ж користувачеві.
- ▶ Не може співпадати з зарезервованими командами Oracle Server.
- ▶ Імена **не залежать** від регістру. Імена в лапках залежать від регістру:

## Однакові

- employees
- EMPLOYEES
- "EMPLOYEES"

## Різні

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

# Створення таблиць: CREATE TABLE

---

► У вас повинні бути:

► Права на операцію **CREATE TABLE**

► Область (зона) зберігання

Ім'я таблиці

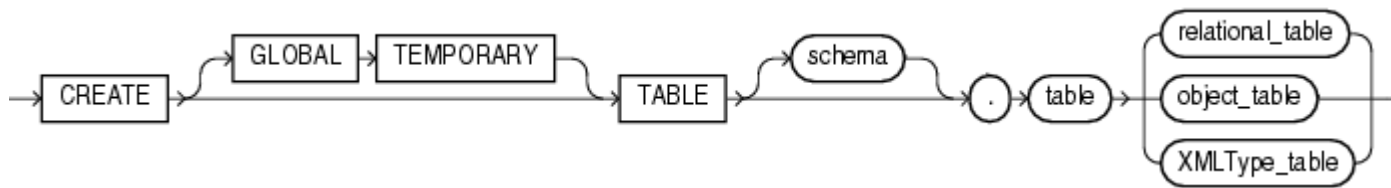
```
CREATE TABLE [schema.]table(  
    column datatype [DEFAULT expr]  
    [,column datatype ...]  
);
```

Ім'я стовпця, його тип і розмір

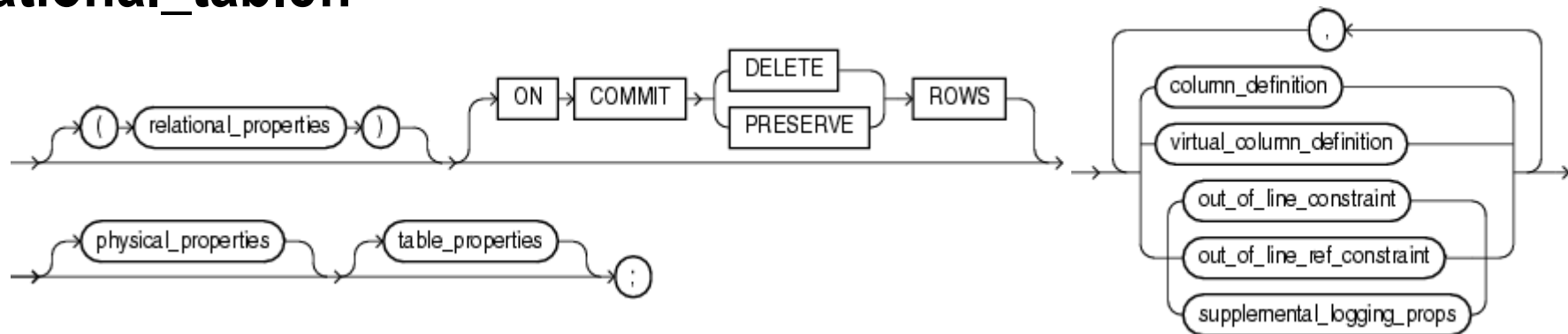
```
CREATE TABLE tbl1(  
    A number,  
    B number DEFAULT 0  
);
```

# Інший спосіб описати допустимий SQL-код

```
CREATE TABLE [schema.] table(  
    column datatype [DEFAULT expr]  
    [, column datatype ...]  
);
```



**relational\_table::=**



[http://docs.oracle.com/html/E26088\\_01/statements\\_7002.htm](http://docs.oracle.com/html/E26088_01/statements_7002.htm)

# Звернення до таблиць інших користувачів

---

- ▶ Таблиці, що належать іншим користувачам, знаходяться поза схемою користувача.
- ▶ Для звернення до них необхідно використовувати ім'я власника в якості префікса в назві таблиці.





# Параметр **DEFAULT**

---

- ▶ Вказує значення за замовчуванням для стовпця при додаванні даних.

```
... hiredate DATE DEFAULT SYSDATE,  
  
City Varchar2(10) DEFAULT 'Sumy' , ...
```

- ▶ Допускаються літерали, вирази і функції SQL.
- ▶ Не допускаються імена інших стовпців або псевдостовпців.
- ▶ Тип даних повинен збігатися з типом даних стовпця.



# Типи даних

---

Тип	Опис
<code>VARCHAR2(size)</code>	Символьні дані змінної довжини
<code>CHAR(size)</code>	Символьні дані фіксованої довжини
<code>NUMBER(p,s)</code>	Чисельні дані змінної довжини
<code>DATE</code>	Дата і час
<code>INTEGER</code>	Ціле число
<code>LONG</code>	Символьні дані змінної довжини до 2 Гб
<code>RAW</code> и <code>LONG RAW</code>	бінарні дані
<code>BLOB</code>	Бінарні дані до 4 Гб
<code>BFILE</code>	Бінарні дані, що зберігаються в зовнішньому файлі, до 4 Гб



## NUMBER [ (кількість цифр [, кількість значущих цифр]) ]

Дані	Тип стовпця	Збережено як
123.89	NUMBER	123.89
123.89	NUMBER(3)	124
123.89	NUMBER(3,2)	Перевихує точність
123.89	NUMBER(4,2)	Перевихує точність
123.89	NUMBER(5,2)	123.89
123.89	NUMBER(6,1)	123.9
123.89	NUMBER(6,-2)	100
.01234	NUMBER(4,5)	.01234
.00012	NUMBER(4,5)	.00012
.000127	NUMBER(4,5)	.00013
.0000012	NUMBER(2,7)	.0000012
.00000123	NUMBER(2,7)	.0000012
1.2e-4	NUMBER(2,5)	0.00012
1.2e-5	NUMBER(2,5)	0.00001

# Приклад

---

```
SQL> CREATE TABLE dept
      2          (deptno NUMBER(2) ,
      3          dname  VARCHAR2(14) ,
      4          loc    VARCHAR2(13)) ;
```

Table created.



# Виведення структури таблиці

Використовуйте команду SQL \* Plus **DESCRIBE** для виведення структури таблиці.

```
DESC[RIBE] tablename
```

```
SQL> DESCRIBE dept
```

Name	Null?	Type
-----		
DEPTNO		NUMBER (2)
DNAME		VARCHAR2 (14)
LOC		VARCHAR2 (13)

# Створення таблиці з використанням підзапиту

---

*Підзапит* – це запит, результат виконання якого, є аргументом іншого запиту. Підзапити називають ще вкладеними запитами.

- ▶ Створення таблиці і додавання рядків за допомогою виразу **CREATE TABLE** і параметра **AS subquery**.

```
CREATE TABLE table  
            [column(, column...)]  
AS subquery;
```

- ▶ Кількість стовпців має збігатися з кількістю стовпців підзапиту.
- ▶ Можна задати імена стовпців та їх значення за замовчуванням.



## Створення таблиці з використанням підзапиту

```
SQL> CREATE TABLE    dept30
  2  AS
  3      SELECT * FROM DEPT;
Table created.
```

```
SQL> DESCRIBE dept30
```

Name	Null?	Type
DEPTNO		NUMBER (2)
DNAME		VARCHAR2 (14)
LOC		VARCHAR2 (13)



# Зміна імені об'єкту

---

- ▶ Щоб змінити ім'я таблиці, представлення, послідовності або синоніма, використовуйте оператор **RENAME**.

```
SQL> RENAME dept TO department;  
Table renamed.
```

- ▶ Ви повинні бути власником об'єкту.



# Очищення таблиці

---

## ▶ Оператор **TRUNCATE TABLE**:

- ▶ Видаляє всі рядки з таблиці;
- ▶ Звільняє місце, зайняте таблицею;
- ▶ Цю дію неможливо відкотити (**Rollback**).

```
SQL> TRUNCATE TABLE department;  
Table truncated.
```

## ▶ Для очищення таблиці також можна використовувати оператор **DELETE**:

```
SQL> DELETE FROM TABLE department;
```



# Знищення таблиці

---

- ▶ Всі дані і структура таблиці знищуються.
- ▶ Усі транзакції підтверджуються.
- ▶ Всі індекси знищуються.
- ▶ Ви *не можете відкотити* цю дію.

```
SQL> DROP TABLE dept30;  
Table dropped.
```



# Оператор ALTER TABLE

---

- ▶ Використовуйте оператор модифікації структури таблиці **ALTER TABLE** для:
  - ▶ Додавання стовпців;
  - ▶ Зміни існуючих стовпців;
  - ▶ Визначення значень за замовчуванням.

```
ALTER TABLE table
ADD           (column datatype [DEFAULT expr]
              [, column datatype]...);
```

```
ALTER TABLE table
MODIFY        (column datatype [DEFAULT expr]
              [, column datatype]...);
```



# Додавання стовпців

DEPT30

EMPNO	ENAME	ANNSAL	HIREDATE
7698	BLAKE	34200	01-MAY-81
7654	MARTIN	15000	28-SEP-81
7499	ALLEN	19200	20-FEB-81
7844	TURNER	18000	08-SEP-81
...			

Новий стовпець

JOB

"... додати  
новий  
стовпець в  
таблицю  
DEPT30 ..."

DEPT30

EMPNO	ENAME	ANNSAL	HIREDATE	JOB
7698	BLAKE	34200	01-MAY-81	
7654	MARTIN	15000	28-SEP-81	
7499	ALLEN	19200	20-FEB-81	
7844	TURNER	18000	08-SEP-81	
...				

# Додавання стовпців

- ▶ Використовуйте специфікатор **ADD** оператора **ALTER TABLE**.

```
SQL> ALTER TABLE dept30  
      2 ADD          (job VARCHAR2(9)) ;  
Table altered.
```

EMPNO	ENAME	ANNSAL	HIREDATE	JOB
7698	BLAKE	34200	01-MAY-81	
7654	MARTIN	15000	28-SEP-81	
7499	ALLEN	19200	20-FEB-81	
7844	TURNER	18000	08-SEP-81	
...				

6 rows selected.

# Змінення стовпця

- ▶ Ви можете змінити тип даних, розмір і значення за замовчуванням для стовпця.

```
ALTER TABLE    dept30
MODIFY          (ename VARCHAR2 (15) ) ;
Table altered.
```

- ▶ Зміна значення за замовчуванням (**DEFAULT**) впливає тільки на наступні дії з таблицею.
- ▶ Зміна типу даних можлива тільки в тому випадку, якщо стовпець порожній;
- ▶ Для незаповненого стовпця можна змінювати розмір/точність. Для заповненого стовпця розмір/точність можна збільшити, але не можна знизити;
- ▶ Обмеження **NOT NULL** може бути встановлене, якщо жодне значення в стовпці не містить **NULL**. Опцію **NOT NULL** завжди можна скасувати;
- ▶ Дозволяється змінювати значення, встановлені за замовчуванням.



# Видалення стовпців

- ▶ Використовуйте специфікатор **DROP** оператора **ALTER TABLE**.

```
ALTER TABLE dept30  
DROP job;
```

EMPNO	ENAME	ANNSAL	HIREDATE
7698	BLAKE	34200	01-MAY-81
7654	MARTIN	15000	28-SEP-81
7499	ALLEN	19200	20-FEB-81
7844	TURNER	18000	08-SEP-81
...			

6 rows selected.

# Додавання коментарів

---

- ▶ Ви можете додати коментар до таблиці або до стовпця за допомогою оператора **COMMENT**.

```
SQL> COMMENT ON TABLE emp  
2 IS 'Employee Information';  
Comment created.
```

- ▶ Коментарі можуть бути переглянуті за допомогою каталогу даних:
  - ▶ **ALL\_COL\_COMMENTS**
  - ▶ **USER\_COL\_COMMENTS**
  - ▶ **ALL\_TAB\_COMMENTS**
  - ▶ **USER\_TAB\_COMMENTS**



# Висновки

---

Вираз	Опис
CREATE TABLE	Створює таблицю
ALTER TABLE	Змінює схему таблиці
DROP TABLE	Знищує таблицю
RENAME	Змінює ім'я таблиці, представлення, послідовності і синоніма
TRUNCATE	Очищає таблицю і звільняє місце, що зайняте таблицею
COMMENT	Додає коментар

