

Лабораторна робота 5

Тема: Фізичне проектування бази даних на СУБД Oracle Database Server 11g ХЕ.

Цель: Дослідити та практично застосувати команди мови SQL в розділі підмови DDL.

Практичні настанови

На основі розробленої даталогічної моделі в лабораторній роботі №3 розробити фізичну модель даних та відповідні скрипти на мові SQL (DDL).

Для створення фізичної моделі необхідно чітко вказати:

- типи даних атрибутів,
- обмеження цілісності,
- спосіб автоматичної генерації значень первинного ключа.

Загальні рекомендації для визначення типів даних для Oracle 11g:

- для коротких символьних значень, чи символьних рядків **фіксованої** довжини – CHAR;
- для символьних рядків **змінної** довжини – VARCHAR з визначенням кількості символів чи максимальним значенням;
- для числових атрибутів NUMBER(n,k), де n – загальна кількість цифр числа, k – кількість цифр після коми. Приклад: щоб зберігати число за шаблоном 999.99, необхідно задати NUMBER(5,2);
- для збереження дат – DATE;
- для збереження великих масивів даних в двійковому чи символьному відображенні – LOB типи;
- для первинних ідентифікаторів – RAW(16) – збереження guid, чи NUMBER для збереження числових ідентифікаторів, враховуючи те, що поле повинно заповнюватись автоматично.

Обмеження цілісності:

- на фізичній моделі позначити первинний ключ, в скриптах створити PRIMARY KEY;
- на фізичній моделі позначити вторинні ключі, в скриптах створити FOREIGN KEY;
- визначити атрибут/атрибути сутностей, які необхідно зробити унікальними, відобразити їх на фізичній моделі, в скриптах створити UNIQUE KEY;
- визначити які поля можуть приймати пусті значення, які – ні та відобразити це на фізичній моделі даних. В скриптах додати властивість NULL\NOT NULL;
- визначити обмеження на введення значень атрибутів та додати в скрипти CHECK.

Спосіб автоматичної генерації первинних ключів:

- для версій Oracle 11g і більш ранніх необхідно створювати SEQUENCE, потім тригер BEFORE INSERT TRIGGER, та використовувати NEXTVAL з послідовності в тригері;

Приклад:

Створення таблиці «Співробітники»

```
CREATE TABLE EMPLOYEES
(ID NUMBER(15) NOT NULL,
SNAME VARCHAR(20) NOT NULL,
FNAME VARCHAR(20) NOT NULL,
PNAME VARCHAR(20) NOT NULL,
GENDER NUMBER(1) DEFAULT 0 NOT NULL,
DATE_BIRTH DATE NULL,
HOME_ADDRESS VARCHAR(100),
PASSPORT VARCHAR(100),
IPN NUMBER(10));
```

Створення послідовності для даної таблиці

```
CREATE SEQUENCE EMPLOYEES_ID_SEQ;
```

Створення тригера

```
CREATE TRIGGER EMPLOYEES_BI
BEFORE INSERT ON EMPLOYEES
FOR EACH ROW
BEGIN
    SELECT EMPLOYEES_ID_SEQ.nextval
    INTO :new.ID
    FROM dual;
END;
```

Відпрацювання автогенерації:

```
INSERT INTO EMPLOYEES (SNAME, FNAME, PNAME) VALUES ('Василенко', 'Сергій', 'Степанович');
COMMIT;
SELECT * FROM EMPLOYEES;
```

	ID	SNAME	FNAME	PNAME	GENDER	DATE_BIRTH	HOME_ADDRESS	PASSPORT	IPN
1	1	Василенко	Сергій	Степанович	0	(null)	(null)	(null)	(null)

- починаючи з версій Oracle 12c, вказувати властивість as IDENTITY первинному ключу при створенні таблиці;

Приклад:

```
SQL> conn C##test/1
Connected.
SQL> CREATE TABLE EMPLOYEES
 2 (ID NUMBER GENERATED BY DEFAULT ON NULL AS IDENTITY,
 3 SNAME VARCHAR(20) NOT NULL,
 4 FNAME VARCHAR(20) NOT NULL,
 5 PNAME VARCHAR(20) NOT NULL,
 6 GENDER NUMBER(1) DEFAULT 0 NOT NULL,
 7 DATE_BIRTH DATE NULL,
 8 HOME_ADDRESS VARCHAR(100),
 9 PASSPORT VARCHAR(100),
10 IPN NUMBER(10));

Table created.
```

Відпрацювання автогенерації:

```
SQL> INSERT INTO EMPLOYEES (SNAME, FNAME, PNAME) VALUES ('Vasylenko', 'Sergiy', 'Steponovych');
1 row created.

SQL> select ID, SNAME, FNAME, PNAME from EMPLOYEES;

      ID SNAME          FNAME          PNAME
-----
      2 Vasylenko      Sergiy        Steponovych

SQL>
```

- при використанні GUID (не дуже часто використовують, так як зберігати та оперувати числами простіше) атрибуту задається тип RAW(16) та у властивості атрибуту при створенні таблиці додається функція SYS_GUID() за замовчуванням.

Приклад:

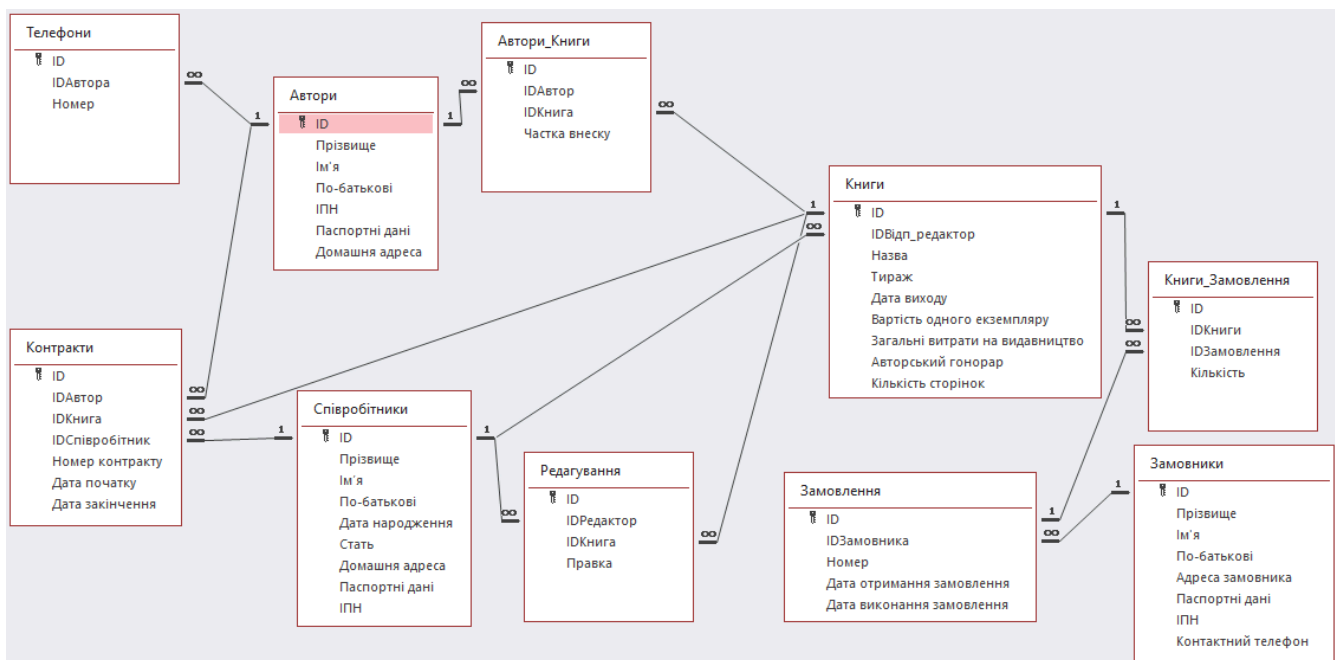
```
CREATE TABLE EMPLOYEES
(ID RAW(16) DEFAULT SYS_GUID(),
SNAME VARCHAR(20) NOT NULL,
FNAME VARCHAR(20) NOT NULL,
PNAME VARCHAR(20) NOT NULL,
GENDER NUMBER(1) DEFAULT 0 NOT NULL,
DATE_BIRTH DATE NULL,
HOME_ADDRESS VARCHAR(100),
PASSPORT VARCHAR(100),
IPN NUMBER(10));
```

Відпрацювання автогенерації:

```
INSERT INTO EMPLOYEES (SNAME, FNAME, PNAME) VALUES ('Василенко', 'Сергій', 'Степанович');
COMMIT;
SELECT * FROM EMPLOYEES;
```

ID	SNAME	FNAME	PNAME	GENDER	DATE_BIRTH	HOME_ADDRESS	PASSPORT	IPN
1 469E370453654CFD938CA8DA4A625C0E	Василенко	Сергій	Степанович	0 (null)	(null)	(null)	(null)	(null)

Даталогічна модель:



Типи даних, що застосовуються в Oracle на сайті офіційного джерела:

QLRF50950

Перелік основних:

CHAR [(size [BYTE | CHAR])]

VARCHAR2(size [BYTE | CHAR])

NVARCHAR2(size)

NUMBER [(p [, s])]

FLOAT [(p)]

DATE

RAW(size)

CLOB

BLOB

Приклад опису сутності з атрибутами певних типів, що застосовуються в СУБД Oracle:

BOOKS		
PK	ID	NUMBER(10) NOT NULL
FK	IDMAINEDITOR	NUMBER(10)
	NAME	VARCHAR2(50) NOT NULL
	CIRCULATION	NUMBER(5)
	DATEPUBLISH	DATE
	COST	NUMBER(5,2)
	EXPENSES	NUMBER(5)
	FEE	NUMBER(5)
	PAGES	NUMBER(4)

AUTHORS		
PK	ID	NUMBER(10) NOT NULL
	SNAME	VARCHAR2(50) NOT NULL
	FNAME	VARCHAR2(50) NOT NULL
	PNAME	VARCHAR2(50)
	DATE_BIRTH	DATE
	PASSPORT	VARCHAR2(50)
	HOMEADDRESS	VARCHAR2(100)

AUTHORS_BOOKS	
ID	NUMBER(10) NOT NULL
IDBOOK	NUMBER(10) NOT NULL
IDAUTHOR	NUMBER(10) NOT NULL
PART	NUMBER(3)

Загальний приклад DDL-команди для створення таблиць:

```
CREATE TABLE table_name
(
  column1 datatype [ NULL | NOT NULL ],
  column2 datatype [ NULL | NOT NULL ],
  ...
  column_n datatype [ NULL | NOT NULL ]
);
```

Розширена інформація:

https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_7002.htm

Створення первинного ключа:

```
ALTER TABLE table_name  
ADD CONSTRAINT constraint_name PRIMARY KEY (column1, column2, ...  
column_n);
```

Створення вторинного ключа:

```
ALTER TABLE table_name  
ADD CONSTRAINT constraint_name  
  FOREIGN KEY (column1, column2, ... column_n)  
  REFERENCES parent_table (column1, column2, ... column_n);
```

Створення послідовності:

```
CREATE SEQUENCE schema_name.sequence_name  
[INCREMENT BY interval]  
[START WITH first_number]  
[MAXVALUE max_value | NOMAXVALUE]  
[MINVALUE min_value | NOMINVALUE]  
[CYCLE | NOCYCLE]  
[CACHE cache_size | NOCACHE]  
[ORDER | NOORDER];
```

Створення

Створення триггеру:

```
CREATE [ OR REPLACE ] TRIGGER trigger_name  
BEFORE INSERT  
ON table_name  
[ FOR EACH ROW ]  
DECLARE  
  -- декларування змінних  
BEGIN  
  -- код триггеру  
EXCEPTION  
WHEN ...  
  -- обробка виключної ситуації  
END;
```

Приклад створення таблиці з ключами та створення послідовності для тригера для автоматичного заповнення первинного ключа:

```
CREATE TABLE BOOKS
(ID number(10) not null,
IDMAINEDITOR number(10),
NAME VARCHAR(20) NOT NULL,
CIRCULATION number(5),
DATEPUBLISH date,
COST number(5,2),
EXPENSES number(5),
FEE number(5),
PAGES number(4));

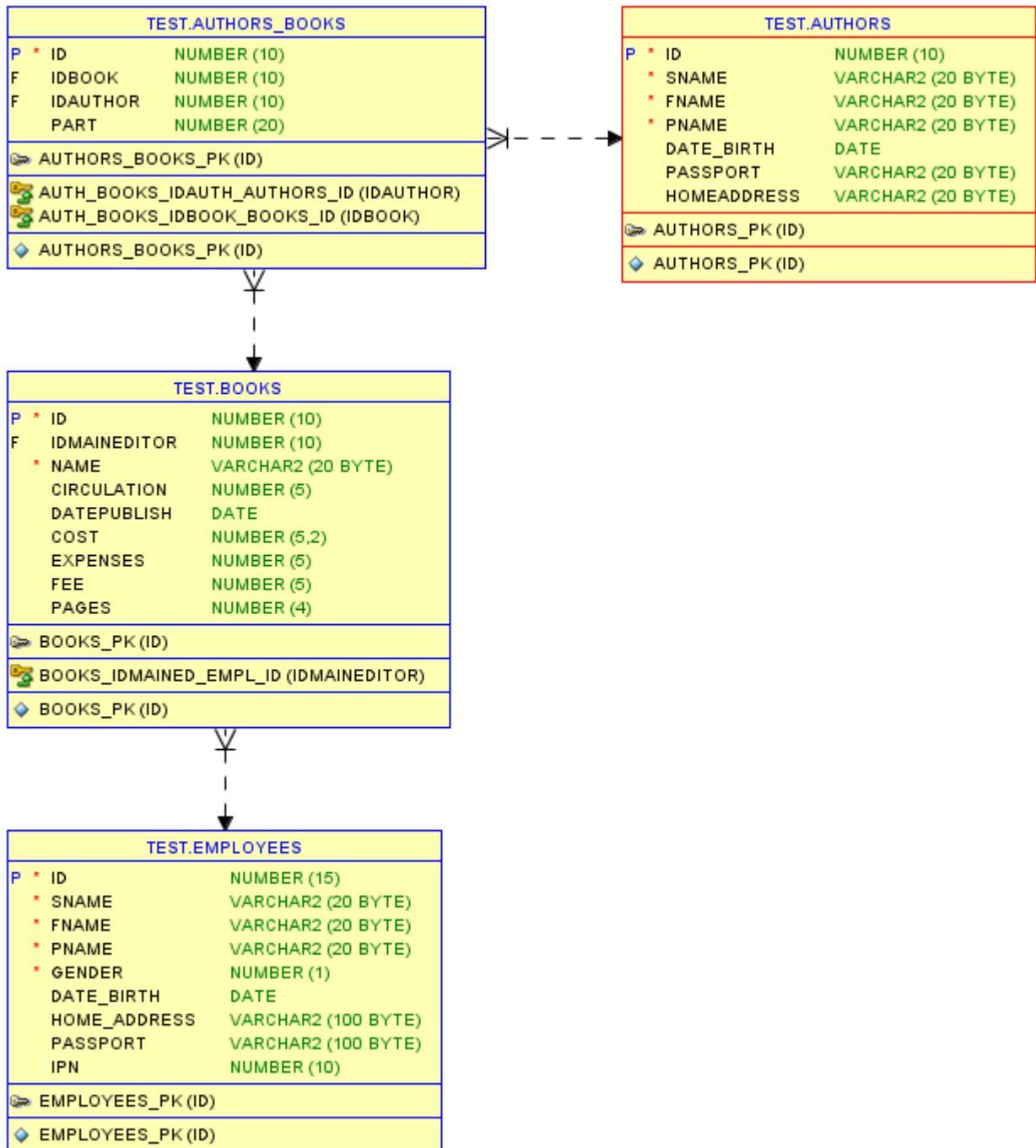
ALTER TABLE BOOKS
ADD CONSTRAINT BOOKS_PK PRIMARY KEY (ID);

ALTER TABLE BOOKS
ADD CONSTRAINT BOOKS_IDMAINED_EMPL_ID
FOREIGN KEY (IDMAINEDEDITOR) REFERENCES EMPLOYEES (ID);

CREATE SEQUENCE BOOKS_ID_SEQ;

CREATE TRIGGER BOOKS_BI
BEFORE INSERT ON BOOKS
FOR EACH ROW
BEGIN
    SELECT BOOKS_ID_SEQ.nextval
    INTO :new.ID
    FROM dual;
END;
```


Фізична модель даних (скорочена):



БЕЗ ФІЗИЧНОЇ РЕАЛІЗАЦІЇ!!!

Дана схема побудована в Oracle SQL Developer. Скрипти для прикладу можна відпрацювати в схемі SYS для побудови моделі. Перед наступною лабораторною таблиці зі схеми SYS слід видалити!!!

Вимоги до виконання лабораторної роботи

1. Шрифт 14, міжрядковий інтервал 1.5.
2. Навести даталогічну модель з лабораторної роботи 3.
3. Навести таблиці з переліком атрибутів та обмежень цілісності для кожної таблиці моделі.
4. Перелік скриптів для створення фізичної схеми даних в СУБД Oracle повинен містити:
 - a. DDL-команди створення таблиць
 - b. DDL-команди створення первинних ключів
 - c. DDL-команди створення вторинних (зовнішніх) ключів
 - d. DDL-команди створення послідовностей
 - e. DDL-команди створення тригерів
 - f. DDL-команди створення унікальних ключів (за необхідності)
 - g. DDL-команди створення обмежень цілісності CHECK (за необхідності)
5. Фізична модель даних з типами даних, обмеженнями цілісності, притаманними певній СУБД (Oracle 11g, 12C, 18C).
6. Скрипти та фізичну модель необхідно будувати повну, для усієї даталогічної моделі.