8 Using DDL Statement to Create and Manage Tables

Lesson Agenda

- Database objects
  - Naming rules
- CREATE TABLE Statement
  - Access another user's tables
  - DEFAULT option
- Data types
- Overview of constraints : `NOT NULL`, `UNIQUE`, `PRIMARY KEY`, `FOREIGN KEY`, `CHECK` constraints
- Creating a table using a subquery
- ALTER TABLE
- DROP TABLE statement

SQL Statements

SELECT INSERT UPDATE DELETE MERGE	Data manipulation language (DML)
CREATE ALTER DROP RENAME TRUNCATE COMMENT	Data definition language (DDL)
GRANT REVOKE	Data control language (DCL)
COMMIT ROLLBACK SAVEPOINT	Transaction control

Database Objects

Object	Description
Table	Basic unit of storage; composed of rows
View	Logically represents subsets of data from one or more tables
Sequence	Generates numeric values
Index	Improves the performance of some queries
Synonym	Gives alternative name to an object

Naming Rules of Tabls

Table names and column names:

- Must begin with a letter
- Must be 1–30 characters long
- Must contain only A–Z, a–z, 0–9, _, $, and #
- Must not duplicate the name of another object owned by the same user
- Must not be an Oracle server–reserved word

CREATE TABLE Statement

Syntax

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

You specify:

- Table name
- Column name, column data type and column size

In the syntax:

- `table` Is the name of the table
- `DEFAULT expr` Specifies a default value if a value is omitted in the INSERT statement
- `column` Is the name of the column
- `Datatype` Is the column's data type and length

DEFAULT Option

- Specify a default value for a column during an insert.

```
... hire_date DATE DEFAULT SYSDATE, ...
```

- Literal values, expressions, or SQL functions are legal values.
- Another column's name or a pseudocolumn are illegal values.
- The default data type must match the column data type.

```
CREATE TABLE hire_dates
        (id         NUMBER(8),
         hire_date DATE  DEFAULT SYSDATE);
```

```
CREATE TABLE succeeded.
```


Data Types

Data Type	Description
VARCHAR2(*size*)	Variable-length character data
CHAR(*size*)	Fixed-length character data
NUMBER(*p*,*s*)	Number having precision p and scale s(Precision is the total number of decimal digits and scale is the number of digits to the right of the decimal point
DATE	Date and time values to the nearest second between January 1, 4712 B.C.,and December 31,9999 A.D.

Including Constraints

- Constraints enforce rules at the table level.
- Constraints prevent the deletion of a table if there are dependencies.
- The following constraint types are valid:
  - NOT NULL
  - UNIQUE
  - PRIMARY KEY
  - FOREIGN KEY
  - CHECK

Including Constraints

Data Integrity Constraints

Constraint	Description
NOT NULL	Specifies that the column cannot contain a null value
UNIQUE	Specifies a column or combination of columns whose values must be unique for all rows in the table
PRIMARY KEY	Uniquely identifies each row of the table
FOREIGN KEY	Establishes and enforces a referential integrity between the column and a column of the referenced table such that values in one table match values in another table.
CHECK	Specifies a condition that must be true

NOT NULL Constraint

Ensures that null values are not permitted for the column:

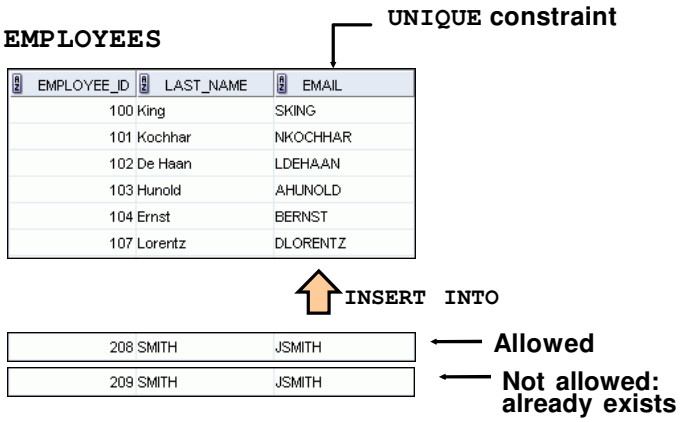
EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	HIRE_DATE	JOB_ID	COMMISSION_PCT
100	Steven	King	SKING	17-JUN-87	AD_PRES	(null)
101	Neena	Kochhar	NKOCHHAR	21-SEP-89	AD_VP	(null)
102	Lex	De Haan	LDEHAAN	13-JAN-93	AD_VP	(null)
103	Alexander	Hunold	AHUNOLD	03-JAN-90	IT_PROG	(null)
104	Bruce	Ernst	BERNST	21-MAY-91	IT_PROG	(null)
107	Diana	Lorentz	DLORENTZ	07-FEB-99	IT_PROG	(null)
124	Kevin	Mourgos	KMOURGOS	16-NOV-99	ST_MAN	(null)
141	Trenna	Rajs	TRAJS	17-OCT-95	ST_CLERK	(null)
142	Curtis	Davies	CDAVIES	29-JAN-97	ST_CLERK	(null)
143	Randall	Matos	RMATOS	15-MAR-98	ST_CLERK	(null)
144	Peter	Vargas	PVARGAS	09-JUL-98	ST_CLERK	(null)
149	Eleni	Zlotkey	EZLOTKEY	29-JAN-00	SA_MAN	0.2
174	Ellen	Abel	EABEL	11-MAY-96	SA_REP	0.3

NOT NULL constraint (Primary Key enforces NOT NULL constraint.)

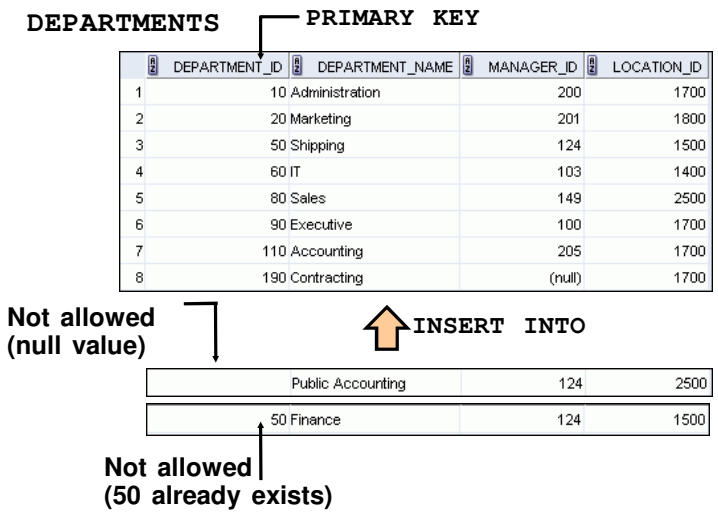
NOT NULL constraint

Absence of NOT NULL constraint (Any row can contain a null value for this column.)

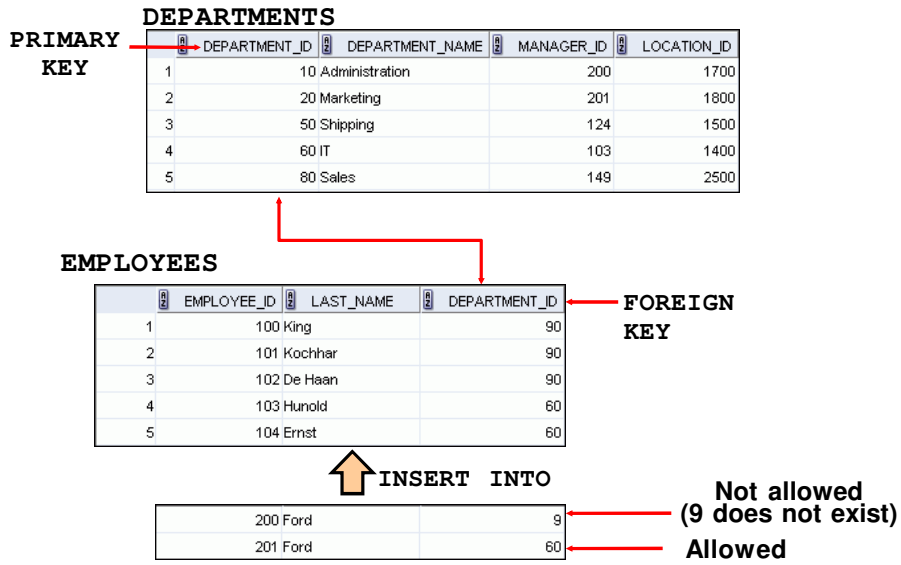
UNIQUE Constraint



PRIMARY KEY Constraint



FOREIGN KEY Constraint



FOREIGN KEY Constraint: Keywords

- **FOREIGN KEY**: Defines the column in the child table at the table-constraint level
- **REFERENCES**: Identifies the table and column in the parent table

CHECK Constraint

- Defines a condition that each row must satisfy
- For example:

```
CREATE TABLE employees
 (……
  salary NUMBER(8,2) CHECK(salary>0),
  …… )
```

การสร้าง table พร้อมกำหนด Constraint

สิ่งที่ควรรู้ก่อนสร้างตาราง:

- Table ที่จะสร้าง ประกอบด้วย column ใดบ้าง?
- แต่ละ column ควรมีข้อมูลแบบใดบ้าง?
- Table นี้ควรมี constraint ใดบ้าง?

การสร้าง table พร้อมกำหนด Constraint

Table CUSTOMER

ชื่อคอลัมน์	ความหมาย	ชนิดข้อมูล	เงื่อนไขหรือข้อกำหนด
CUST_NO	รหัสประจำตัวลูกค้า	NUMBER(5)	primary key
NAME	ชื่อลูกค้า	CHAR(20)	not null
ADDRESS	ที่อยู่ลูกค้า	VARCHAR2(40)	not null
DOB	วันเดือนปีเกิดลูกค้า	DATE	
ID_CARD_NO	เลขที่บัตรประชาชนลูกค้า	NUMBER(13)	unique
CUST_TYPE	ประเภทของลูกค้า	CHAR(1)	มีได้สามค่าคือ A=ลูกค้าชั้นดีเยี่ยม B=ลูกค้าชั้นดี C=ลูกค้าปกติ

การสร้าง table พร้อมกำหนด Constraint

```
CREATE TABLE  customer
(cust_no    NUMBER(5)      PRIMARY KEY,
 name       CHAR(20)       NOT NULL,
 address    VARCHAR2(40)   NOT NULL,
 dob        DATE,
 id_card_no NUMBER(13)     UNIQUE,
 cust_type  CHAR(1)
            CHECK(cust_type IN('A','B','C'))
);
```


การสร้าง table พร้อมกำหนด Constraint

```
CREATE TABLE  customer1
(cust_no    NUMBER(5)  ,
 name       CHAR(20)       NOT NULL,
 address    VARCHAR2(40)   NOT NULL,
 dob        DATE,
 id_card_no NUMBER(13),
 cust_type  CHAR(1),

 PRIMARY KEY(cust_no),
 UNIQUE(id_card_no),
 CHECK(cust_type IN('A','B','C')));
```

not null ต้องระบุแบบ column constraint เท่านั้น

การระบุ constraint แบบ table constraint จะเขียนหลังจากเขียนคำสั่งในส่วนที่เกี่ยวกับ column จบ

การสร้าง table พร้อมระบุ Foreign key constraint

Table ORDER1

ชื่อคอลัมน์	ความหมาย	ชนิดข้อมูล	เงื่อนไขหรือข้อกำหนด
ORD_NO	หมายเลข ORDER	NUMBER(5)	primary key
ORD_DATE	วันที่บันทึก ORDER	DATE	Not null, Default เป็น system date
AMOUNT	ยอดเงินรวมต่อ ORDER	NUMBER(9,2)	not null
CUST_NO	รหัสประจำตัวลูกค้าที่สั่ง ORDER	NUMBER(5)	foreign key อ้างไปยัง CUST_NO ในtable ชื่อ CUSTOMER

การสร้าง table พร้อมระบุ Foreign key constraint

Practice # 1

Create an Oracle table called suppliers that stores supplier ID, name, and address information.

Solution for Practice Exercise #1:

- The Oracle CREATE TABLE statement for the suppliers table is:

Practice # 2

Create an Oracle table called customers that stores customer ID, customer name, and age information. But this time, the ID should be the primary key for the table.

Solution for Practice Exercise #2:

- The Oracle CREATE TABLE statement for the customers table is:

Practice # 3

- Create on the depts table as below,
- Create an Oracle table called emps that stores employee number, employee name, department id, and salary information.
- The primary key for the emps table should be the employee number. And create a foreign key on the emps table that references the depts table based on the dept_id field.

Practice # 3

Solution for Practice Exercise #3:

- The Oracle CREATE TABLE statement for the emps table is:

Creating a Table Using a Subquery

- Create a table and insert rows by combining the `CREATE TABLE` statement and the AS `subquery` option.
- Match the number of specified columns to the number of subquery columns.
- Define columns with column names and default values.

Creating a Table Using a Subquery

Syntax:

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

SQL Statement:

```
CREATE  TABLE  dept80
AS   SELECT   employee_id,  last_name,
             salary,hire_date
    FROM     employees
    WHERE    department_id =  80;
```

Creating a Table Using a Subquery

SQL Statement:

```
DESCRIBE dept80;
```

```
Name                          Null     Type
----------------------------- -------- -------------
EMPLOYEE_ID                            NUMBER(6)
LAST_NAME                     NOT NULL VARCHAR2(25)
ANNSAL                                 NUMBER
HIRE_DATE                     NOT NULL DATE
```

```
SELECT * FROM dept80;
```


Practice

Create the SALES_REPS table based on the structure of the EMPLOYEES table. Include only the EMPLOYEE_ID, FIRST_NAME, SALARY, and COMMISSION_PCT columns. Name the columns in your new table ID, NAME, SALARY, and COMMISSION where employees are sales representative.

ALTER TABLE Statement

Use the ALTER TABLE statement to:

- Add a new column
- Modify an existing column definition
- Define a default value for the new column
- Drop a column
- Rename a column
- Change table to read-only status

Add column in table

Syntax: To ADD A COLUMN in a table, the Oracle ALTER TABLE syntax is:

```
ALTER TABLE   table-name
ADD  (column_name    column-definition);
```

Example:

- Add column fname is char size 30 to table dept80

```
ALTER TABLE dept80
ADD  (fname  char(30));
```

```
desc dept80;
```

Add multiple columns in table

Syntax: To ADD MULTIPLE COLUMNS to an existing table, the Oracle ALTER TABLE syntax is:

```
ALTER TABLE   table-name
ADD (column_1    column-definition,
     column_2    column-definition,
     ……….
     column_n   column-definition
);
```

Add column address and salary to *customers* table.

```
ALTER TABLE customers
  ADD (address    varchar2(50),
       salary     number(10,2)
);
```


Modify column in table

Syntax: To MODIFY A COLUMN in an existing table, the Oracle ALTER TABLE syntax is:

```
ALTER TABLE table-name
MODIFY (column_name column_datatype);
```

Example:

- Modify column fname size is 50 from table dept80

```
ALTER TABLE dept80
MODIFY (surname   char(30));
```

```
desc dept80;
```


Drop column in table

Syntax: To DROP A COLUMN in an existing table, the Oracle ALTER TABLE syntax is:

```
ALTER TABLE table-name
DROP COLUMN column_name;
```

Example:

- Delete column fname from table dept80

```
ALTER TABLE dept80
DROP COLUMN fname;
```

```
desc dept80;
```


Rename column in table

Syntax: To RENAME A COLUMN in an existing table, the Oracle ALTER TABLE syntax is:

```
ALTER TABLE table_name
RENAME COLUMN old_name to new_name;
```

Example:

- Rename column customer_id to id and column customer_name to name from table customers.

```
ALTER TABLE  customers
RENAME  COLUMN  customer_id  to  id;
```

```
ALTER TABLE  customers
RENAME  COLUMN  customer_name  to name;
```

```
desc customers;
```

Rename table

Syntax: To RENAME A TABLE, the Oracle ALTER TABLE syntax is:

```
ALTER TABLE table_name
RENAME TO new_table_name;
```

Example:

- Rename a table customers to contacts

```
ALTER TABLE customers RENAME TO contacts;
```


ADD CONSTRAINT PRIMARY KEY

Syntax:

```
ALTER TABLE table-name
ADD CONSTRAINT CONSTRAINT_NAME PRIMARY KEY(COLUMN);
```

Add a table-level PRIMARY KEY constraint to the SALES_REPS table on the EMPLOYEE_ID column. The constraint should be named at creation. Name the constraint emp_id_pk.

```
ALTER TABLE sales_reps
ADD CONSTRAINT emp_id_pk PRIMARY KEY(id);
```


ADD CONSTRAINT FOREIGN KEY

Syntax:

```
ALTER TABLE table-name
ADD CONSTRAINT CONSTRAINT_NAME
FOREIGN KEY COLUMN REFERENCES TABLE(COLUMN);
```

Add a FOREIGN KEY constraint the SALES_REPS table on the ID column to Depts table on the DEP_ID column. The constraint should be named at creation. Name the constraint emp_dep_id_fk.

```
ALTER TABLE sales_reps
ADD CONSTRAINT emp_dept_id_fk
FOREIGN KEY(id) REFERENCES depts(dep_id);
```


DROP Table

Syntax: To DROP A Table in an existing database, the Oracle DROP TABLE syntax is:

```
DROP TABLE table-name;
```

Example:

- Delete table order1 from database.

```
DROP TABLE order1;
```

```
Delete table customer,customer1,order2 from
database.
```

Practice # 1

Based on the depts table below, rename the deps table to department.

```
CREATE TABLE depts
( dept_id    number(10) not null,
  dept_name  varchar2(50) not null,
  PRIMARY KEY(dept_id)
);
```

Solution for Practice Exercise # 1:

- The following Oracle ALTER TABLE statement would rename the depts table to department:

Practice # 2

Based on the emps table below, add a column called bonus that is a number(6) datatype.

```
CREATE TABLE emps
( employee_number  number(10) not null,
  employee_name    varchar2(50) not null,
  dept_id    number(10),
  PRIMARY KEY(employee_number)
);
```

Solution for Practice Exercise #2:

- The following Oracle ALTER TABLE statement would add a bonus column to the emps table:

Practice # 3

Based on the contacts table below, add one column called last_contacted that is a date datatype.

Practice # 3

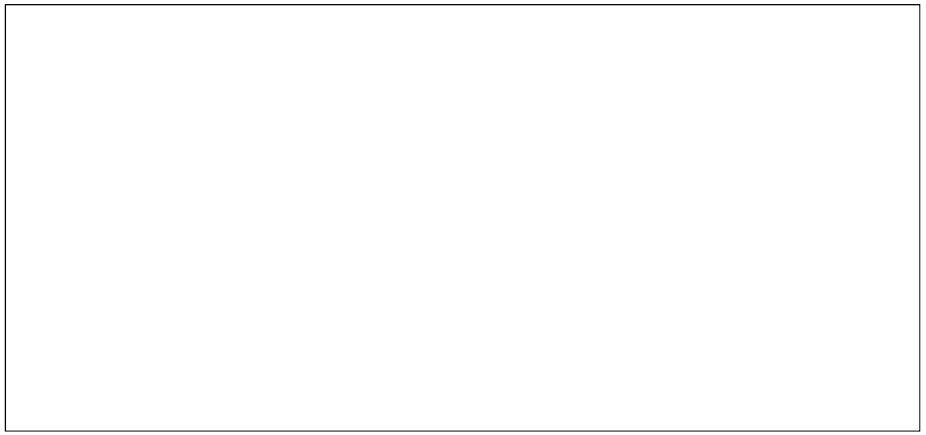
Solution for Practice Exercise #3:

- The following Oracle ALTER TABLE statement would add the last_contacted columns to the contacts table:

Practice # 4

Based on the emps table below, change the emp_name column to a varchar2(75) datatype.

Solution for Practice Exercise #4:

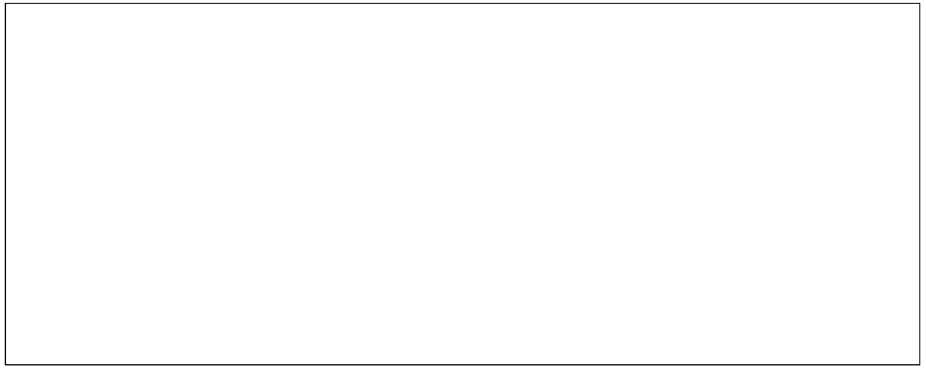


Practice # 5

Based on the contacts table below, drop the last_contacted column.

Solution for Practice Exercise #5:

- The following Oracle ALTER TABLE statement would drop the last_contacted column from the contacts table:



Practice # 6

Based on the department table below, rename the dept_name column to department_name.

Solution for Practice Exercise #6:

- The following Oracle ALTER TABLE statement would rename the dept_name column to department_name in the department table:

Summary

In this lesson, you should have learned how to use the CREATE TABLE statement to create a table and include constraints:

- Categorize the main database objects
- Review the table structure
- List the data types that are available for columns
- Create a simple table
- Explain how constraints are created at the time of table creation