
Lab Week – 5

Creating Tables and Using Constraints

Objectives

- Creating tables:
 - Describe the main database objects
 - Create tables
 - Describe the data types that can be used when specifying column definition
 - Alter table definitions
 - Drop, rename, and truncate tables
- Using Constraints
 - Describe constraints
 - Create and maintain constraints

Database Objects

Object	Description
Table	Basic unit of storage; composed of rows and columns
View	Logically represents subsets of data from one or more tables
Sequence	Generates primary key values
Index	Improves the performance of some queries
Synonym	Gives alternative names to objects

Naming Conventions

- Must begin with a letter
- Can 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

The CREATE TABLE Statement

- You must have :
 - CREATE TABLE privilege
 - A storage area

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

- You specify:
 - Table name
 - Column name, column data type, and column size

Creating Tables

- Create the table.

```
CREATE TABLE dept
  (deptno      NUMBER(2),
   dname       VARCHAR2(14),
   loc         VARCHAR2(13),
   create_date DATE DEFAULT SYSDATE);
```

Table created.

- Confirm table creation.

```
DESCRIBE dept
```

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

Data Types

Data Type	Description
CHAR(size)	Fixed-length character data
VARCHAR2	Variable-length character data
CLOB	Character data up to 8 Terabytes
NUMBER(p,s)	Variable-length numeric data
DATE	Date and Time values
BLOB	Binary data up to 8 Terabytes
RAW & LONG RAW	Raw binary data
BFILE	Binary data stored in operating-system files outside the database; up to 8 Terabytes

The **DEFAULT** Option

- Specify a default value for a column, to be used during an insert.
- Legal values are literal value, expression, or SQL function.
- The default datatype must match the column datatype.

```
... hiredate DATE DEFAULT SYSDATE, ...
```

Creating a table by using a sub-query

- Create a table and insert rows by combining the CREATE TABLE statement and AS subquery option.

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

- Match the number of specified columns to the number of subquery columns.
- Define columns with column names and default values.

Creating a table by using a sub-query

```
SQL> CREATE TABLE    dept30
  2  AS
  3      SELECT      empno, ename, sal*12 ANNSAL, hiredate
  4      FROM        emp
  5      WHERE       deptno = 30;
```

Table created.

```
SQL> DESCRIBE dept30
```

Name	Null?	Type
-----	-----	-----
EMPNO	NOT NULL	NUMBER(4)
ENAME		VARCHAR2(10)
ANNSAL		NUMBER
HIREDATE		DATE

Tables in the Oracle Database

- User Tables
 - Collection of tables created and maintained by the user
 - Contain user information
- Data Dictionary
 - Collection of tables created and maintained by the Oracle server
 - Contain database information

Querying the Data Dictionary

- List tables owned by the user.

```
SQL> SELECT    table_name  
  2  FROM      user_tables;
```

- View distinct object types owned by the user.

```
SQL> SELECT    DISTINCT object_type  
  2  FROM      user_objects;
```

- View tables, views, synonyms, and sequences owned by the user.

```
SQL> SELECT    *  
  2  FROM      user_catalog;
```

Referencing Another User's Tables

- Tables belonging to other users are not in the user's schema.
- You should use the owner's name as a prefix to the table.

```
SELECT *
FROM tsg0781.Customer;
```

The ALTER TABLE Statement

- Use the ALTER TABLE statement to:
 - Add a new column
 - Modify an existing column
 - Define a default value for the new column

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

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

Adding a Column

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	
...				

New column

“...add a new column into DEPT30 table...”

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	
...				

Adding a Column

- You use the ADD clause to add columns.

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

- The new column becomes the last column.

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.				

Modifying a Column

- You can change a column's data type, size, and default value.

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

- A change to the default value affects only subsequent insertions to the table.

Dropping a Table

- All data and structure in the table is deleted.
- Any pending transactions are committed.
- All indexes are dropped.
- You can not roll back this statement.

```
SQL> DROP TABLE dept30;
```

Table dropped.

Changing the Name of an Object

- To change the name of a table, view, sequence, or synonym, you execute the RENAME statement.

```
SQL> RENAME dept TO my_department;
```

Table renamed.

- You must be the owner of the object.

Truncating a Table

- The TRUNCATE TABLE statement:
- Removes all rows from a table
- Releases the storage space used by that table

```
SQL> TRUNCATE TABLE dept;
```

Table truncated.

- You can not roll back row removal when using TRUNCATE.
- Alternatively, you can remove rows by using the DELETE statement – this will be covered next week.

Adding Comments to a Table

- You can add comments to a table or column by using the COMMENT statement.

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

- Comments can be viewed through the data dictionary views.
 - ALL_COL_COMMENTS
 - USER_COL_COMMENTS
 - ALL_TAB_COMMENTS
 - USER_TAB_COMMENTS

What Are 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 in Oracle:
 - NOT NULL
 - UNIQUE
 - PRIMARY KEY
 - FOREIGN KEY
 - CHECK

Constraint Guidelines

- Name a constraint or the Oracle Server will generate a name by using the SYS_Cn format.
- Create a constraint:
 - At the same time as the table is created
 - After the table has been created
- Define a constraint at the column or table level.
- View a constraint in the data dictionary.

Defining Constraints

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

```
CREATE TABLE emp(  
    empno  NUMBER(4),  
    ename  VARCHAR2(10),  
    ...  
    deptno NUMBER(7,2) NOT NULL,  
    CONSTRAINT emp_empno_pk  
        PRIMARY KEY (EMPNO));
```

Defining Constraints

- Column constraint level

```
column [CONSTRAINT constraint_name] constraint_type,
```

- Table constraint level

```
column, . . .
[CONSTRAINT constraint_name] constraint_type
(column, . . .),
```

The NOT NULL Constraint

- Ensures that null values are not permitted for the column

EMP

EMPNO	ENAME	JOB	...	COMM	DEPTNO
7839	KING	PRESIDENT			10
7698	BLAKE	MANAGER			30
7782	CLARK	MANAGER			10
7566	JONES	MANAGER			20
...					

NOT NULL constraint
(no row can contain
a null value for
this column)

Absence of NOT NULL
constraint
(any row can contain
null for this column)

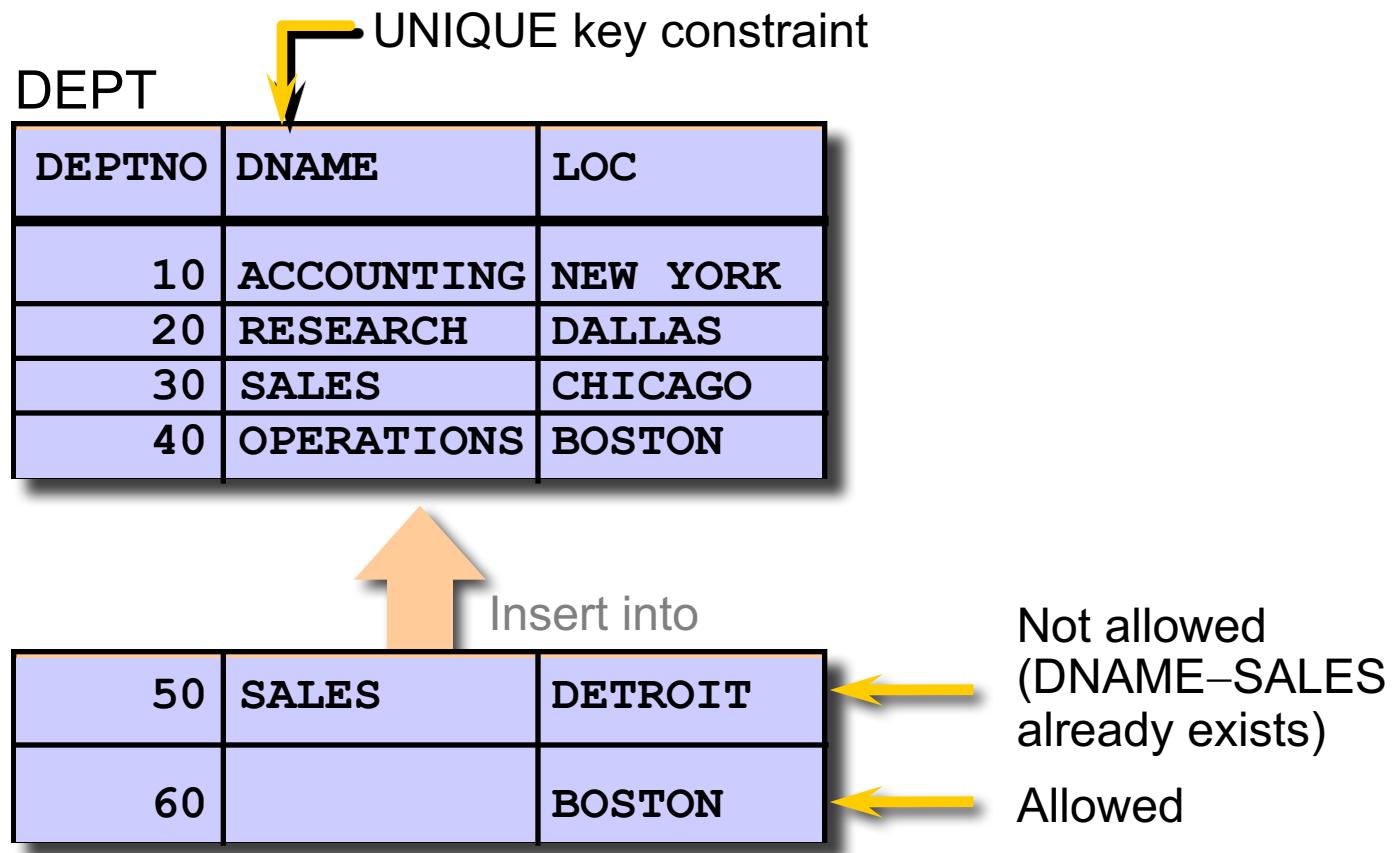
NOT NULL constraint

The NOT NULL Constraint

- Defined at the column level

```
SQL> CREATE TABLE emp (
  2      empno    NUMBER(4),
  3      ename     VARCHAR2(10) NOT NULL,
  4      job       VARCHAR2(9),
  5      mgr       NUMBER(4),
  6      hiredate  DATE,
  7      sal        NUMBER(7,2),
  8      comm      NUMBER(7,2),
  9      deptno    NUMBER(7,2) NOT NULL);
```

The UNIQUE Key Constraint



The UNIQUE Key Constraint

- May be defined at either the table level or the column level
- Defined at table level:

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

The PRIMARY KEY Constraint

DEPT

PRIMARY KEY

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

Insert into

20	MARKETING	DALLAS
	FINANCE	NEW YORK

Not allowed
(DEPTNO=20 already exists)

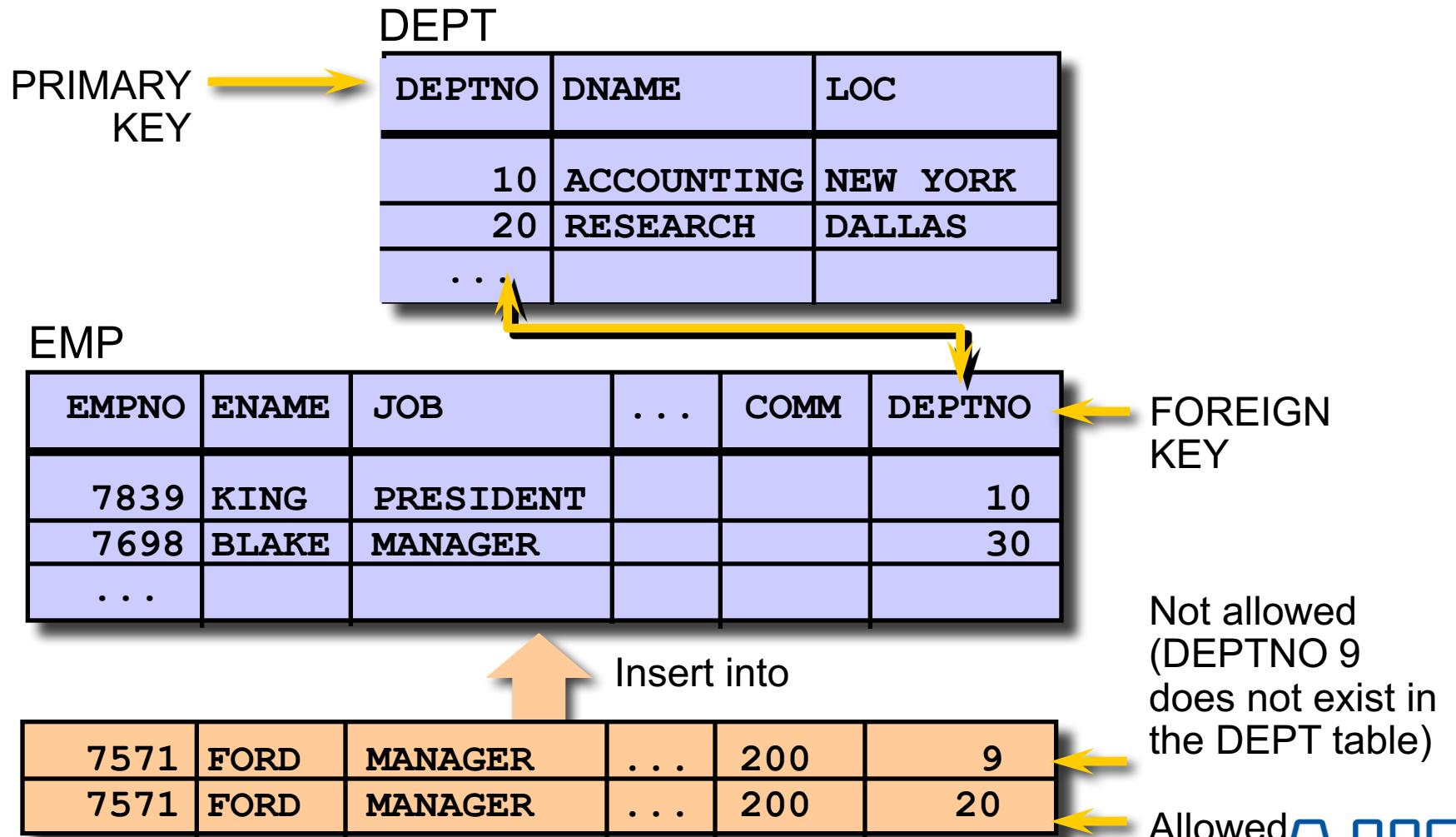
Not allowed
(DEPTNO is null)

The PRIMARY KEY Constraint

- Defined at either the table level or the column level

```
SQL> CREATE TABLE dept(
  2      deptno      NUMBER(2),
  3      dname       VARCHAR2(14),
  4      loc         VARCHAR2(13),
  5      CONSTRAINT dept_dname_uk UNIQUE (dname),
  6      CONSTRAINT dept_deptno_pk PRIMARY KEY(deptno)) ;
```

The FOREIGN KEY Constraint



The FOREIGN KEY Constraint

- Defined at either the table level or the column level

```
SQL> CREATE TABLE emp (
  2      empno      NUMBER(4),
  3      ename      VARCHAR2(10) NOT NULL,
  4      job        VARCHAR2(9),
  5      mgr        NUMBER(4),
  6      hiredate   DATE,
  7      sal         NUMBER(7,2),
  8      comm        NUMBER(7,2),
  9      deptno     NUMBER(7,2) NOT NULL,
 10      CONSTRAINT emp_deptno_fk FOREIGN KEY (deptno)
 11          REFERENCES dept (deptno));
```

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
- ON DELETE CASCADE
 - Allows deletion in the parent table and deletion of the dependent rows in the child table
- Other ON DELETE Options
 - ON DELETE {CASCADE | NO ACTION | SET DEFAULT}

The CHECK Constraint

- Defines a condition that each row must satisfy
- Expressions that are not allowed:
 - References to CURRVAL, NEXTVAL, LEVEL, and ROWNUM pseudocolumns
 - Calls to SYSDATE, UID, USER, and USERENV functions
 - Queries that refer to other values in other rows

```
..., deptno NUMBER(2),  
      CONSTRAINT emp_deptno_ck  
          CHECK (DEPTNO BETWEEN 10 AND 99),...
```

Adding a Constraint

- Add or drop, but not modify, a constraint
- Enable or disable constraints
- Add a NOT NULL constraint by using the MODIFY clause

```
ALTER TABLE table
ADD [CONSTRAINT constraint] type (column);
```

Adding a Constraint

- Add a FOREIGN KEY constraint to the EMP table indicating that a manager must already exist as a valid employee in the EMP table.

```
SQL> ALTER TABLE      emp
  2 ADD CONSTRAINT   emp_mgr_fk
  3          FOREIGN KEY(mgr) REFERENCES emp(empno) ;
Table altered.
```

Dropping a Constraint

- Remove a foreign key constraint from the EMP table.

```
SQL> ALTER TABLE      emp
  2  DROP CONSTRAINT  emp_mgr_fk;
Table altered.
```

- Remove the PRIMARY KEY constraint on the DEPT table and drop the associated FOREIGN KEY constraint on the EMP.DEPTNO column.

```
SQL> ALTER TABLE      dept
  2  DROP PRIMARY KEY CASCADE;
Table altered.
```

Disabling Constraints

- Execute the DISABLE clause of the ALTER TABLE statement to deactivate an integrity constraint.
- Apply the CASCADE option to disable dependent integrity constraints.

```
SQL> ALTER TABLE emp  
  2  DISABLE CONSTRAINT emp_empno_pk CASCADE;  
Table altered.
```

Enabling Constraints

- Activate an integrity constraint currently disabled in the table definition by using the ENABLE clause.

```
SQL> ALTER TABLE emp  
  2  ENABLE CONSTRAINT emp_empno_pk;  
Table altered.
```

- A UNIQUE or PRIMARY KEY index is automatically created if you enable a UNIQUE key or PRIMARY KEY constraint.

Viewing Constraints

- Query the USER_CONSTRAINTS table to view all constraint definitions and names.

```
SQL> SELECT constraint_name, constraint_type,  
2          search_condition  
3     FROM user_constraints  
4    WHERE table_name = 'EMP' ;
```

CONSTRAINT_NAME	C SEARCH_CONDITION
SYS_C00674	C EMPNO IS NOT NULL
SYS_C00675	C DEPTNO IS NOT NULL
EMP_EMPNO_PK	P
...	

Viewing the Columns Associated with Constraints

- View the columns associated with the constraint names in the USER_CONS_COLUMNS view.

```
SQL> SELECT constraint_name, column_name  
2   FROM user_cons_columns  
3  WHERE table_name = 'EMP';
```

CONSTRAINT_NAME	COLUMN_NAME
EMP_DEPTNO_FK	DEPTNO
EMP_EMPNO_PK	EMPNO
EMP_MGR_FK	MGR
SYS_C00674	EMPNO
SYS_C00675	DEPTNO

Summary

- Create table, Alter table, Drop table
- Create the following types of constraints:
 - NOT NULL
 - UNIQUE
 - PRIMARY KEY
 - FOREIGN KEY
 - CHECK
- Query the USER_CONSTRAINTS table to view all constraint definitions and names.

Lab Activities

- Complete exercise handed out in lab