Constraints

Objectives

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

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 Key
  - PRIMARY KEY
  - FOREIGN KEY
  - CHECK

Constraint Guidelines

- Name a constraint or the Oracle Server will generate a name by using the SYS_C*n* 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 data type [DEFAULT expr]
             [column_constraint],
             …
             [table_constraint]);
```

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

Defining Constraints

- Column constraint level

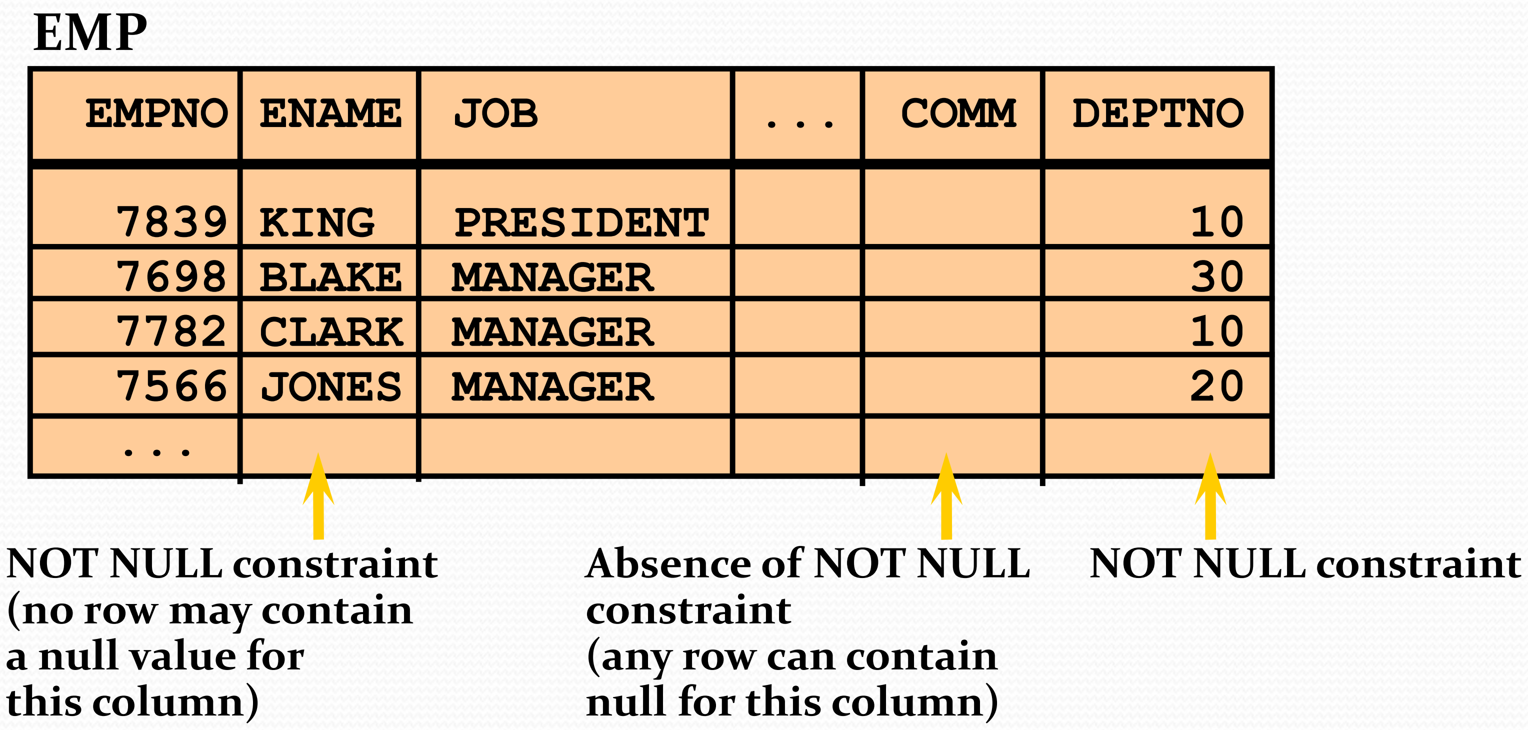
```
column [CONSTRAINT constraint_name] constraint_type,
```

- 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 may contain
a null value for
this column)

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

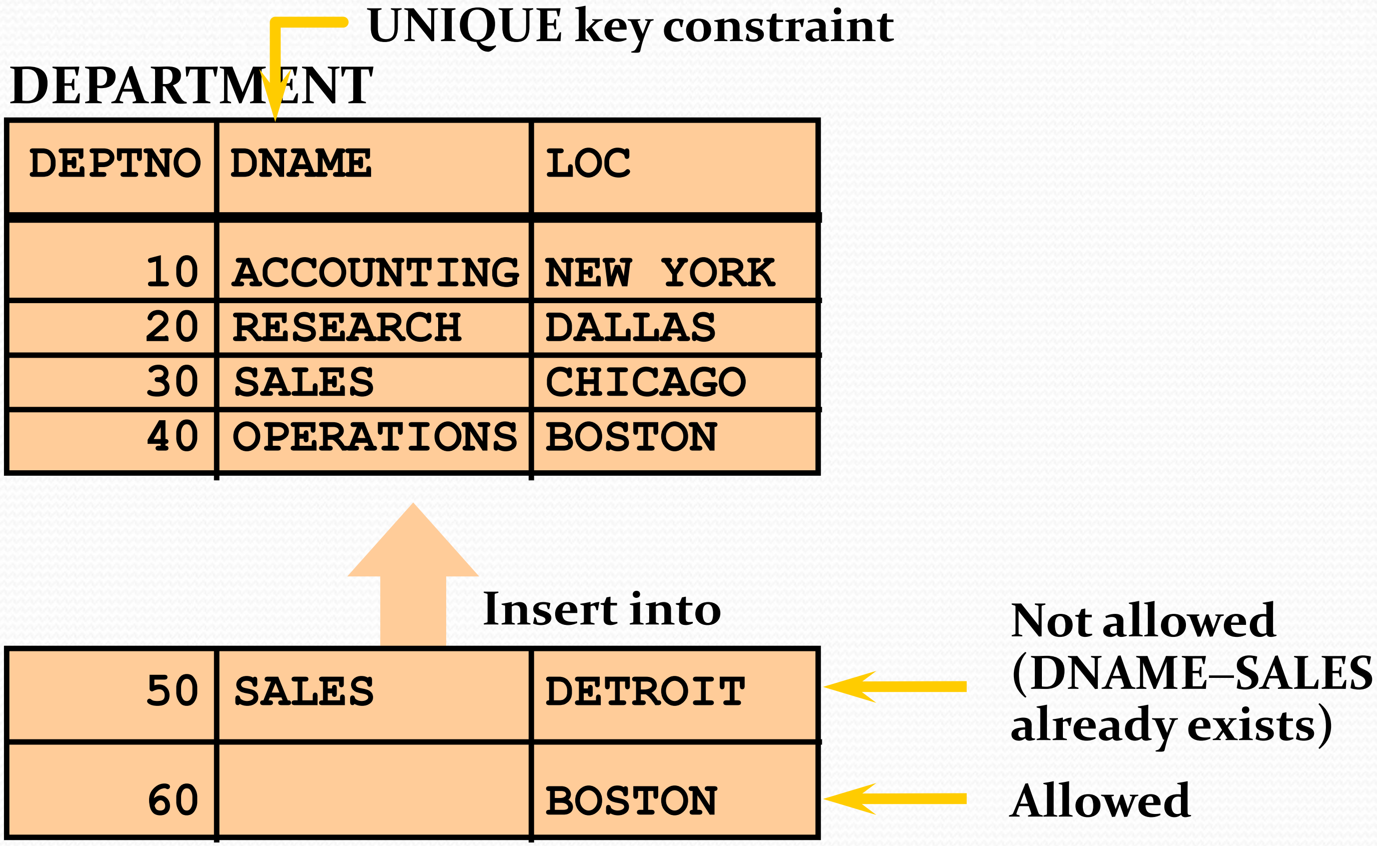
NOT NULL constraint

The NOT NULL Constraint

- Defined at the column level

```
SQL> CREATE TABLE employee(
  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



UNIQUE key constraint

DEPARTMENT

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

Insert into

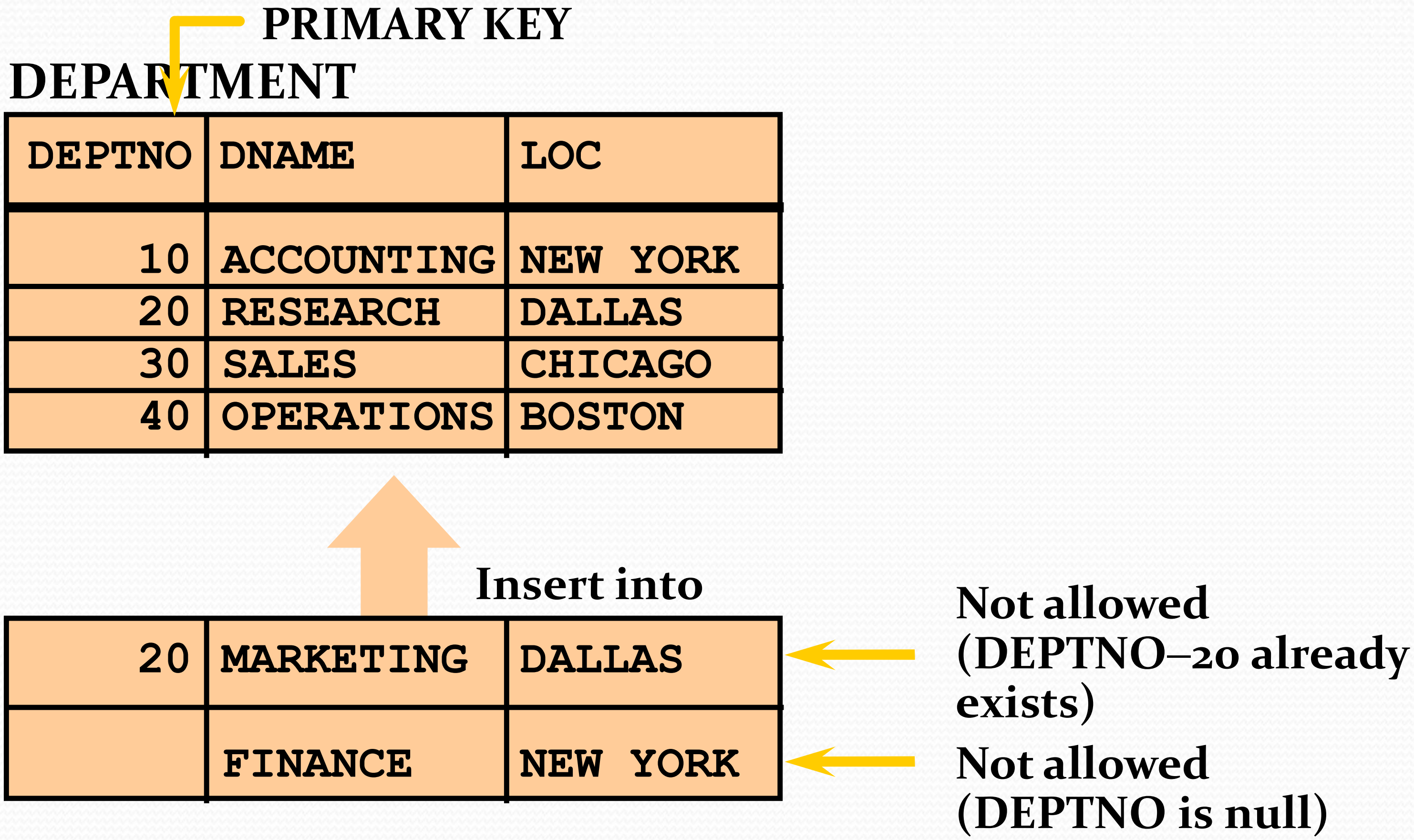
50	SALES	DETROIT	Not allowed (DNAME–SALES already exists)
60		BOSTON	Allowed

The UNIQUE Key Constraint

- Defined at either the table level or the column level

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

PRIMARY KEY Constraint



PRIMARY KEY

DEPARTMENT

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)

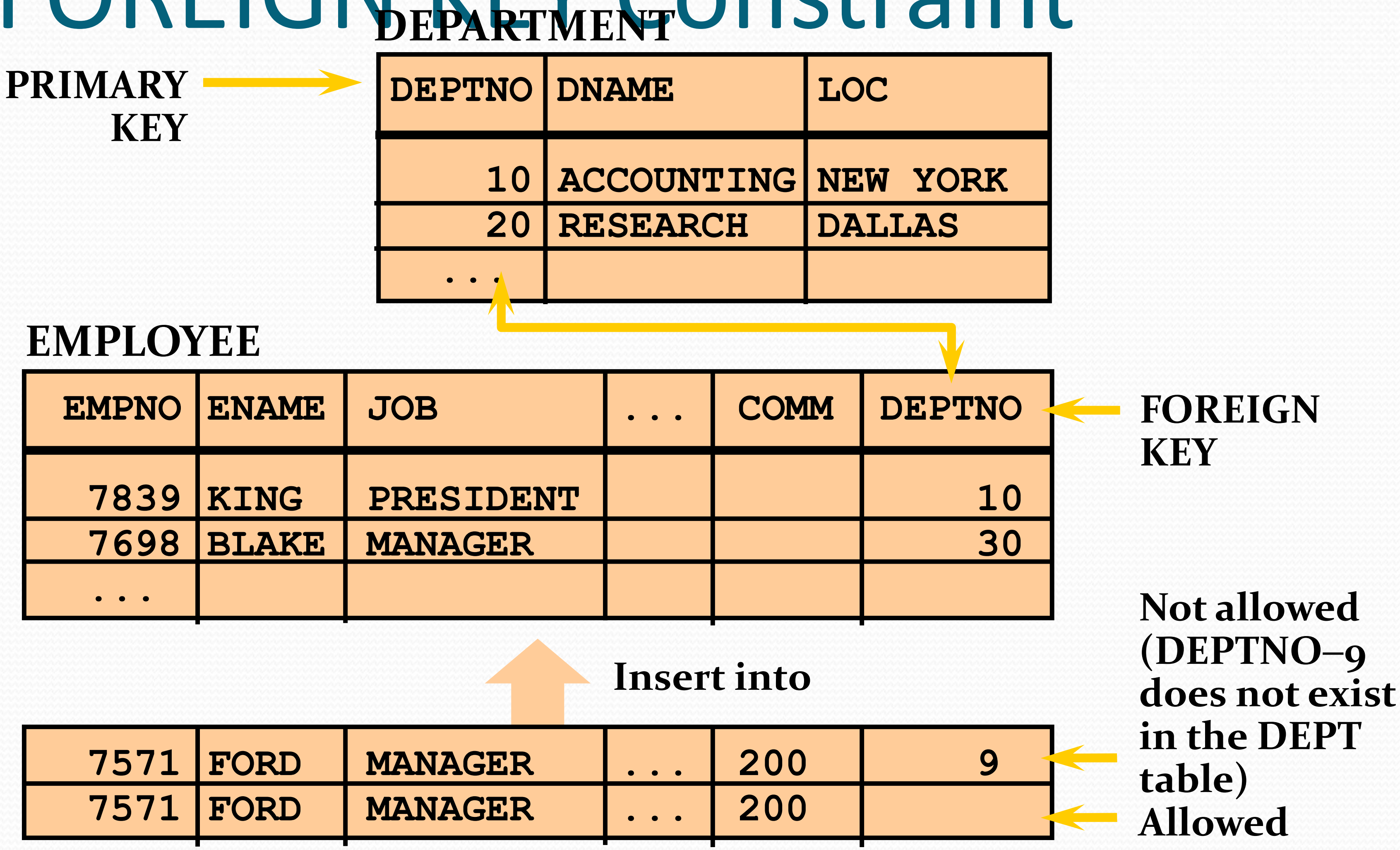
Not allowed (DEPTNO is null)

PRIMARY KEY Constraint

- Defined at either the table level or the column level

```
SQL> CREATE TABLE   department(
  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));
```

FOREIGN KEY Constraint



DEPARTMENT

PRIMARY KEY

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
...		

EMPLOYEE

EMPNO	ENAME	JOB	...	COMM	DEPTNO
7839	KING	PRESIDENT			10
7698	BLAKE	MANAGER			30
...					

FOREIGN KEY

Insert into

7571	FORD	MANAGER	...	200	9
7571	FORD	MANAGER	...	200	

Not allowed (DEPTNO–9 does not exist in the DEPT table)

Allowed

FOREIGN KEY Constraint

- Defined at either the table level or the column level

```
SQL> CREATE TABLE employee(
  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

The CHECK Constraint

- Defines a condition that each row must satisfy
- Expressions that are not allowed:
  - References to pseudo columns CURRVAL, NEXTVAL, LEVEL, and ROWNUM
  - 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

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

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

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     employee
  2  ADD CONSTRAINT  emp_mgr_fk
  3            FOREIGN KEY(mgr) REFERENCES emp(empno);
Table altered.
```

Dropping a Constraint

- Remove the manager constraint from the EMP table.

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

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 = 'EMPLOYEE';
```

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

Columns 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 = 'EMPLOYEE';
```

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

Summary

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