

ORACLE®

# Constraints

## 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:
  - NOT NULL
  - UNIQUE
  - PRIMARY KEY
  - FOREIGN KEY
  - CHECK



# 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 foreign key relationship between the column and a column of the referenced table
CHECK	Specifies a condition that must be true

# Constraints

## Defining Constraints

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

## Constraints

```
CREATE TABLE emp90
```

```
(  
  emp_id CHAR(6), first_name CHAR(10), job_id CHAR(5) NOT NULL,  
  CONSTRAINT emp_id_pk PRIMARY KEY (emp_id)  
)
```

### Defining Constraints

- Column constraint level

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

- Table constraint level

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

# Constraints

## The NOT NULL Constraint

Ensures that null values are not permitted for the column:

EMPLOYEE_ID	LAST_NAME	FNAME	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	DEPARTMENT_ID
100	King	SKING	515.123.4567	17-JUN-87	AD_PRES	24000	90
101	Kochhar	NKOCHHAR	515.123.4568	21-SEP-89	AD_VP	17000	90
102	De Haan	LDEHAAN	515.123.4569	13-JAN-93	AD_VP	17000	90
103	Hunold	AHUNOLD	690.423.4567	03-JAN-90	IT_PROG	9000	60
104	Ernst	BERNST	690.423.4568	21-MAY-91	IT_PROG	6000	60
170	Grant	KGRANT	011.44.1644.429263	24-MAY-99	SA_REP	7000	
200	Whalen	JWHALEN	515.123.4444	17-SEP-87	AD_ASST	4400	10

...  
20 rows selected.

↑  
**NOT NULL constraint**  
(No row can contain  
a null value for  
this column.)

↑  
**NOT NULL  
constraint**

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

# Constraints

## The NOT NULL Constraint

Is defined at the column level:

```
CREATE TABLE employees(  
  employee_id    NUMBER(6),  
  last_name      VARCHAR2(25) NOT NULL,  
  salary         NUMBER(8,2),  
  commission_pct NUMBER(2,2),  
  hire_date      DATE  
                  CONSTRAINT emp_hire_date_nn  
                  NOT NULL,  
  ...  
)
```

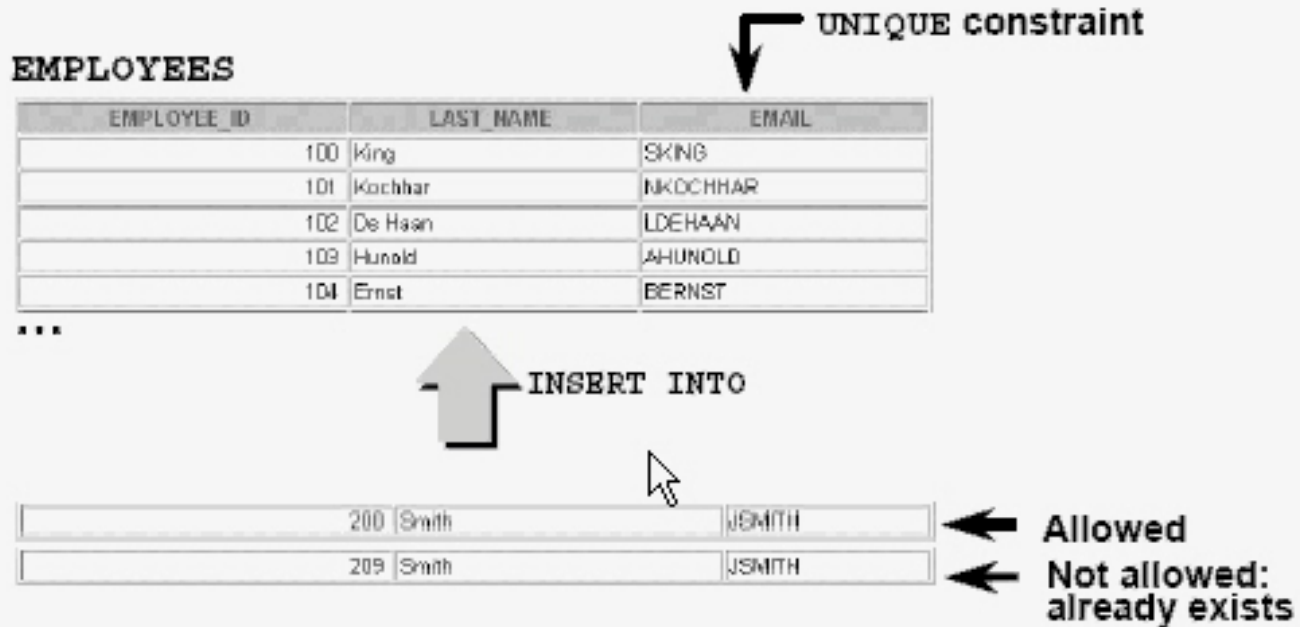
← System named

← User named

Constraints at column or table level could be hold user named else it have system named

# Constraints

## The UNIQUE Constraint






# Constraints

## The UNIQUE Constraint

Defined at either the table level or the column level:

```
CREATE TABLE employees(  
    employee_id      NUMBER(6),  
    last_name        VARCHAR2(25) NOT NULL,  
    email            VARCHAR2(25),  
    salary           NUMBER(8,2),  
    commission_pct   NUMBER(2,2),  
    hire_date        DATE NOT NULL,  
    ...  
    CONSTRAINT emp_email_uk UNIQUE(email));
```



# Constraints

## The PRIMARY KEY Constraint

DEPARTMENTS



PRIMARY KEY

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1800
60	IT	103	1400
80	Sales	149	2500

...

Not allowed  
(Null value)



INSERT INTO

	Public Accounting		1400
50	Finance	124	1800

Not allowed  
(50 already exists)



# Constraints

## The PRIMARY KEY Constraint

Defined at either the table level or the column level:

```
CREATE TABLE departments(  
    department_id      NUMBER(4),  
    department_name     VARCHAR2(30)  
        CONSTRAINT dept_name_nn NOT NULL,  
    manager_id         NUMBER(6),  
    location_id         NUMBER(4),  
        CONSTRAINT dept_id_pk PRIMARY KEY(department_id));
```



# Constraints

## The FOREIGN KEY Constraint

### DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500

PRIMARY  
KEY →

...

### EMPLOYEES

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
100	King	90
101	Kochhar	90
102	De Haan	90
103	Hunold	60
104	Ernst	60
107	Lorentz	60

← FOREIGN  
KEY

...

↑ INSERT INTO

200	Ford	9
201	Ford	60

← Not allowed  
(9 does not  
exist)

← Allowed

ORACLE

# Fk and Pk in same table

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PC	MANAGER_ID	DEPARTMENT_ID
100	Steven	King	SKING	215.123.4567	17-JUN-87	AD_PRES	250	-		90
101	Neena	Kochhar	NKOCHHAR	215.123.4568	21-SEP-89	AD_VP	17000	-	100	90
102	Lex	De Haan	LDEHAAN	215.123.4569	13-JAN-93	AD_VP	17000	-	100	90
103	Alexander	Hunold	AHUNOLD	590.423.4567	03-JAN-90	IT_PROG	9000	-	102	60
104	Bruce	Ernst	BERNST	590.423.4568	21-MAY-91	IT_PROG	6000	-	103	60
105	David	Austin	DAUSTIN	590.423.4569	25-JUN-97	IT_PROG	4800	-	103	60
106	Valli	Pataballa	VPATABAL	590.423.4560	05-FEB-98	IT_PROG	4800	-	103	60
107	Diana	Lorentz	DLORENTZ	590.423.5567	07-FEB-99	IT_PROG	4200	-	103	60
108	Nancy	Greenberg	NGREENBE	215.124.4569	17-AUG-94	FI_MGR	12000	-	101	100
109	Daniel	Faviet	DFAVIET	215.124.4169	16-AUG-94	FI_ACCOUNT	9000	-	108	100

More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.03 seconds

CSV Export

# Constraints

## The FOREIGN KEY Constraint

Defined at either the table level or the column level:

```
CREATE TABLE employees(  
    employee_id      NUMBER(6),  
    last_name        VARCHAR2(25) NOT NULL,  
    email            VARCHAR2(25),  
    salary            NUMBER(8,2),  
    commission_pct   NUMBER(2,2),  
    hire_date        DATE NOT NULL,  
    ...  
    department_id    NUMBER(4),  
    CONSTRAINT emp_dept_fk FOREIGN KEY (department_id)  
        REFERENCES departments(department_id),  
    CONSTRAINT emp_email_uk UNIQUE(email));
```

# Constraints

## **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:** Deletes the dependent rows in the child table when a row in the parent table is deleted.
- **ON DELETE SET NULL:** Converts dependent foreign key values to null

# Constraints

## The CHECK Constraint

- Defines a condition that each row must satisfy
- The following expressions 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

```
..., salary  NUMBER(2)  
          CONSTRAINT emp_salary_min  
             CHECK (salary > 0),...
```





# Constraints

## Adding a Constraint Syntax

Use the **ALTER TABLE** statement to:

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

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

- CREATE TABLE suppliers
- ( supplier\_id numeric(4),
- supplier\_name varchar2(50),
- CONSTRAINT check\_supplier\_id CHECK (supplier\_id BETWEEN 100 and 9999) )
- alter table suppliers add(CONSTRAINT s\_name CHECK (supplier\_name = upper(supplier\_name)))
- alter table suppliers add(CONSTRAINT s\_name\_con CHECK (supplier\_name IN('ALI','AZAD')))
- alter table suppliers add(CONSTRAINT s\_age CHECK (age < 5))
- insert into suppliers values(120,'AZAD',5)
- update suppliers set age=null
- truncate table suppliers
- select \* from suppliers

<http://www.techonthenet.com/oracle/index.php>

# Constraints

## Adding a Constraint

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

```
ALTER TABLE      employees
ADD CONSTRAINT    emp_manager_fk
    FOREIGN KEY(manager_id)
    REFERENCES employees(employee_id);
Table altered.
```

# Constraints

## Syntax

```
ALTER TABLE table
DROP PRIMARY KEY | UNIQUE (column) |
    CONSTRAINT constraint [CASCADE];
```

In the syntax:

<i>table</i>	is the name of the table
<i>column</i>	is the name of the column affected by the constraint
<i>constraint</i>	is the name of the constraint

# Constraints

## Dropping a Constraint

- Remove the manager constraint from the **EMPLOYEES** table.

```
ALTER TABLE      employees
DROP CONSTRAINT    emp_manager_fk;
Table altered.
```

- Remove the **PRIMARY KEY** constraint on the **DEPARTMENTS** table and drop the associated **FOREIGN KEY** constraint on the **EMPLOYEES.DEPARTMENT\_ID** column.

```
ALTER TABLE      departments
DROP PRIMARY KEY CASCADE;
Table altered.
```

# Constraints

## 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.

```
ALTER TABLE      employees
DISABLE CONSTRAINT emp_emp_id_pk CASCADE;
Table altered.
```

# Constraints

## Enabling Constraints

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

```
ALTER TABLE      employees
ENABLE CONSTRAINT  emp_emp_id_pk;
Table altered.
```

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

# Constraints

```
CREATE TABLE test1 (  
  pk NUMBER PRIMARY KEY,  
  fk NUMBER,  
  col1 NUMBER,  
  col2 NUMBER,  
  CONSTRAINT fk_constraint FOREIGN KEY (fk) REFERENCES test1,  
  CONSTRAINT ck1 CHECK (pk > 0 and col1 > 0),  
  CONSTRAINT ck2 CHECK (col2 > 0));
```

An error is returned for the following statements:

```
ALTER TABLE test1 DROP (pk);
```

-- pk is a parent key

```
ALTER TABLE test1 DROP (col1);
```

-- col1 is referenced by multicolumn constraint ck1

Statements

Errors returned



# Constraints

## Cascading Constraints

Example:

```
ALTER TABLE test1  
DROP (pk) CASCADE CONSTRAINTS;  
Table altered.
```

```
ALTER TABLE test1  
DROP (pk, fk, coll) CASCADE CONSTRAINTS;  
Table altered.
```



## Constraints

```
SELECT constraint_name,constraint_type,search_condition  
FROM user_constraints  
WHERE table_name='EMPLOYEES'
```

CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION
EMP_LAST_NAME_NN	C	"LAST_NAME" IS NOT NULL
EMP_EMAIL_NN	C	"EMAIL" IS NOT NULL
EMP_HIRE_DATE_NN	C	"HIRE_DATE" IS NOT NULL
EMP_JOB_NN	C	"JOB_ID" IS NOT NULL
EMP_SALARY_MIN	C	salary > 0
EMP_EMAIL_UK	U	-
EMP_EMP_ID_PK	P	-
EMP_DEPT_FK	R	-
EMP_JOB_FK	R	-
EMP_MANAGER_FK	R	-

Constraint type C check, U unique, P primary and R foreign

## Constraints

```
SELECT constraint_name,column_name  
FROM user_cons_columns  
WHERE table_name='EMPLOYEES'
```

CONSTRAINT_NAME	COLUMN_NAME
EMP_EMAIL_UK	EMAIL
EMP_SALARY_MIN	SALARY
EMP_JOB_NN	JOB_ID
EMP_HIRE_DATE_NN	HIRE_DATE
EMP_EMAIL_NN	EMAIL
EMP_LAST_NAME_NN	LAST_NAME
EMP_MANAGER_FK	MANAGER_ID
EMP_JOB_FK	JOB_ID
EMP_DEPT_FK	DEPARTMENT_ID
EMP_EMP_ID_PK	EMPLOYEE_ID