

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)

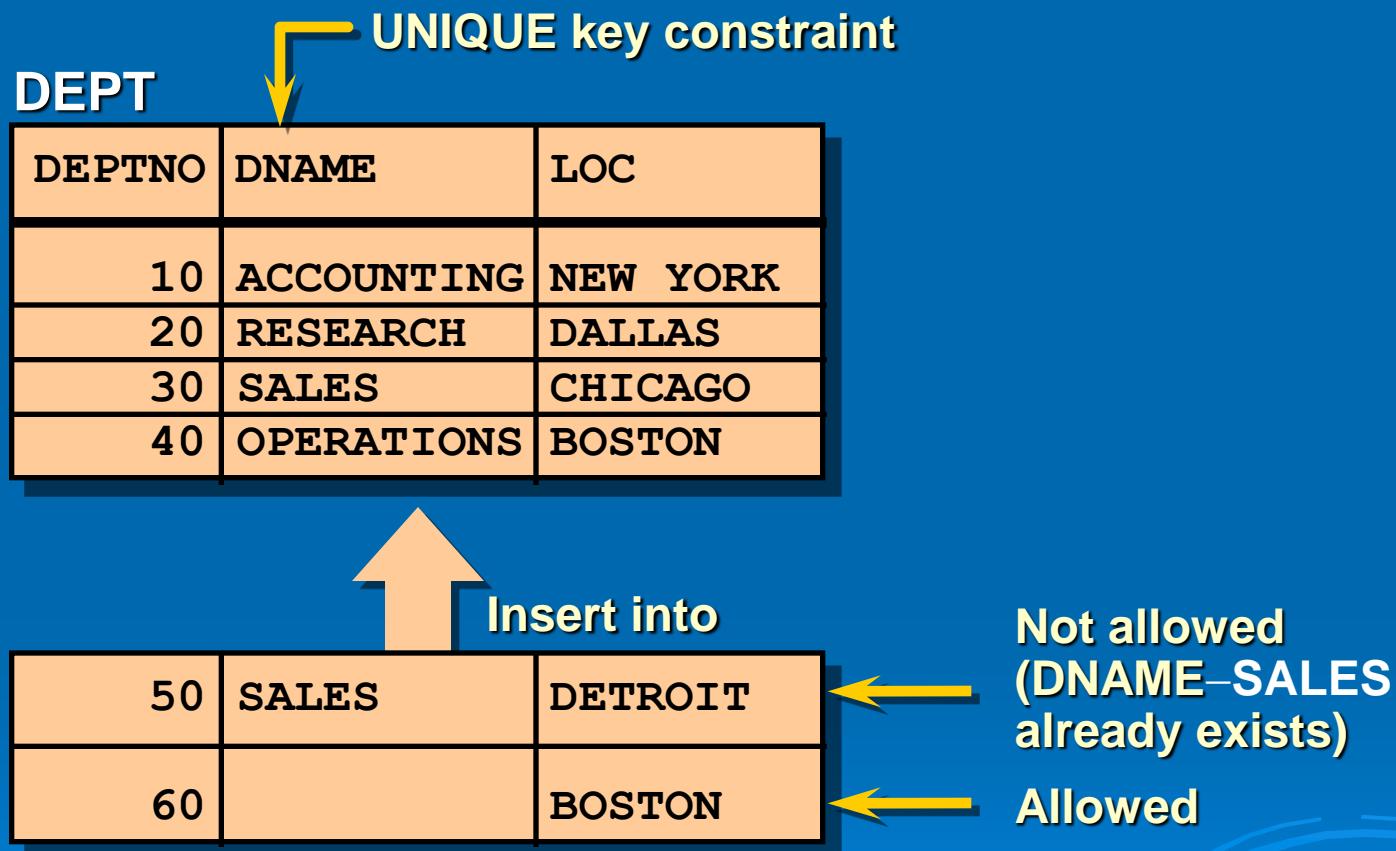
Absence of NOT NULL
constraint
(any row can contain
null for this column)

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

- 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) );
```

The PRIMARY KEY Constraint

DEPT 

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)

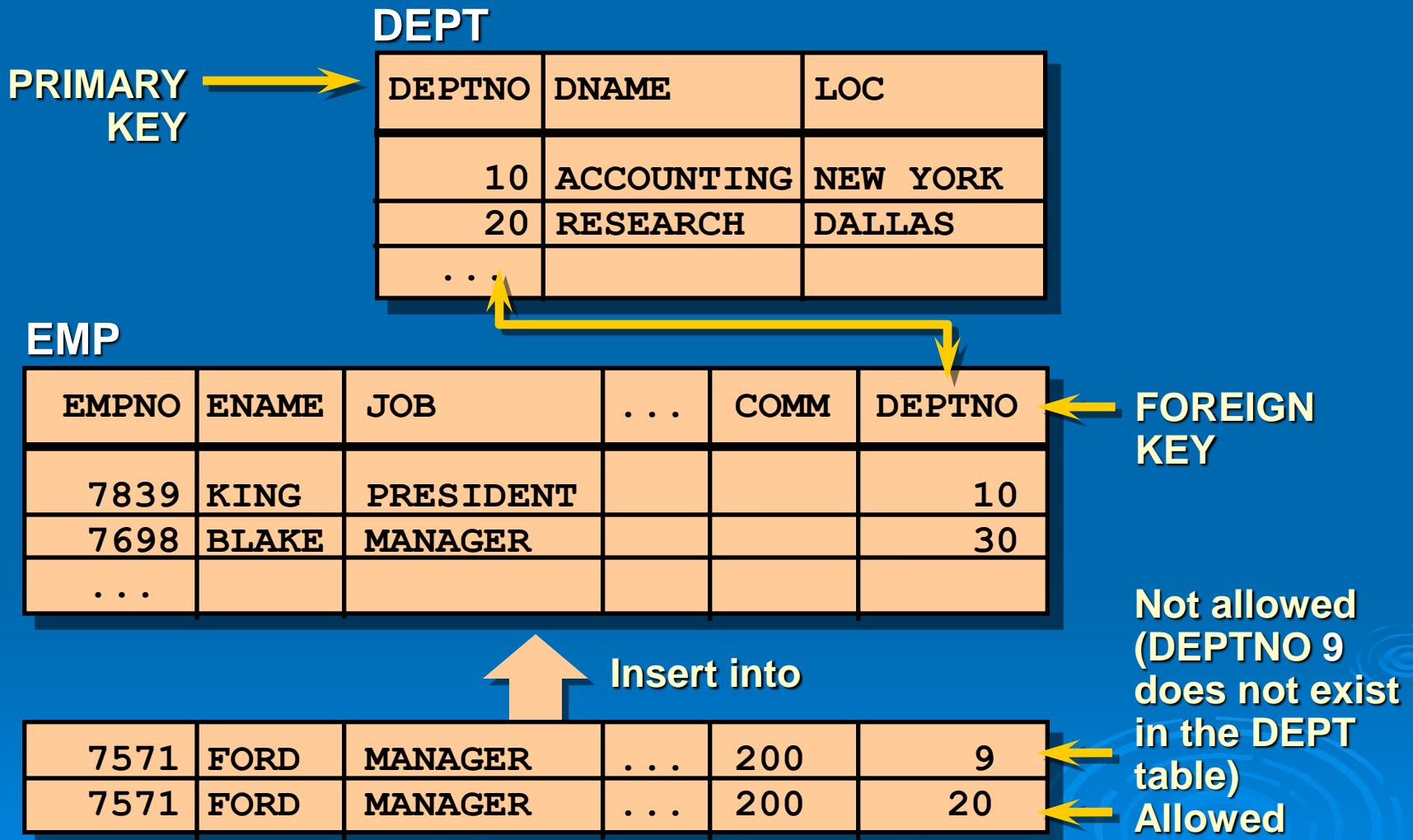
Not allowed
(DEPTNO is null)

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

```
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

The CHECK Constraint

- Defines a condition that each row must satisfy

```
..., 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      emp
  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          emp
  2  DROP CONSTRAINT    emp_mgr_fk;
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.

ALTER TABLE Examples

- ALTER TABLE student
ADD (address VARCHAR(40))
- ALTER TABLE student
MODIFY (lname VARCHAR2(35))
- ALTER TABLE student
DROP COLUMN gpa

Views



Defining Views

- A view can be referred to as a “virtual table”
- A view is created by using the statement

CREATE VIEW

OUR TABLE

Lets use the following table for an example:

F_NAME	L_NAME	AGE	JOB	SALARY
JOHN	MEZA	51	CONSULTANT	45000
AISHA	KHALID	21	ACCOUNTANT	35000
MERIAM	KHAN	25	ASSISTANT	19000
MANSOOR	KHAN	53	MANAGER	57000
MEHMOOD	SYED	50	ASSISTANT M	50000
ASIYA	JAVED	35	CONSULTANT	43000

Why use Views

- Views restrict access to the database
- Makes complex queries easier
- A View contains no data of its own
- Can create a view using multiple tables
- Users have access to particular data according to their needs

An example

```
CREATE VIEW EMPLOYEE_INFO
```

```
SELECT * FROM EMPLOYEE;
```

F_NAME	L_NAME	AGE	JOB	SALARY
JOHN	MEZA	51	CONSULTANT	45000
AISHA	KHALID	21	ACCOUNTANT	35000
MERIAM	KHAN	25	ASSISTANT	19000
MANSOOR	KHAN	53	MANAGER	57000
MEHMOOD	SYED	50	ASSISTANT M	50000
ASIYA	JAVED	35	CONSULTANT	43000

Creating Views from Views

- A view can be made from a previous view.
- There is a downfall of this. When you have a view a few levels down from the original view, when the previous level is dropped from the database the following will be useless due to them receiving their information from the previous view.

Example of a view from a view

```
CREATE VIEW OVER_40 AS  
SELECT F_NAME, L_NAME  
FROM EMPLOYEE_INFO  
WHERE AGE>40;
```

<u>F_NAME</u>	<u>L_NAME</u>	<u>AGE</u>	<u>JOB</u>	<u>SALARY</u>
JOHN	MEZA	51	CONSULTANT	45000
MANSOOR	KHAN	53	MANAGER	57000
MEHMOOD	SYED	50	ASSISTANT M	50000

Renaming columns in Views

- Another advantage of using views is that you can change the names of the columns to display as you wish.
- For example if you wanted to concatenate two columns to appear as one column in your view.

Example

```
Create view EMPLOYEE_JOB  
(FULL_NAME , JOB) AS  
SELECT F_NAME + “ “ + L_NAME, JOB  
FROM EMPLOYEE_INFO;
```

Updating information in views

- Say we wanted to see what our employees will be making with a 5% raise

```
CREATE VIEW RAISE  
SELECT * FROM EMPLOYEE INFO;
```

Then:

```
UPDATE RAISE  
SET SALARY = SALARY + SALARY * .05;
```

Denying DML Operations

- DML operation could be denied by using the READ ONLY statement

CREATE VIEW...

AS SELECT

.....

WITH READ ONLY;

Removing a view

- Remove a view without losing data
- `DROP VIEW v_name;`