Chapter 4
Structure Query Language

SQL - Constraints

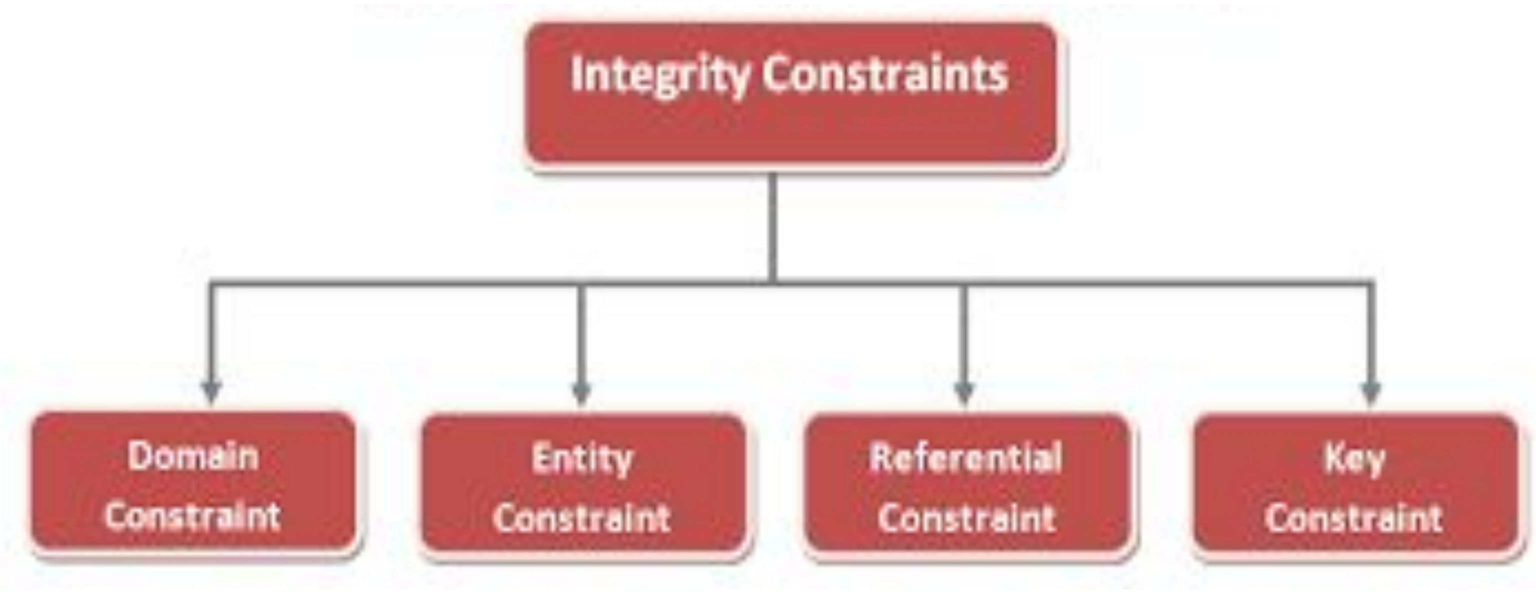
Department: Computer
Course: DBMS
Faculty: Sana Shaikh

What is Constraints ?

- ✔ Maintaining integrity of a database is the most important factor.
- ✔ Some limitations have to be enforced on the data, and only that data which satisfies the conditions will actually be stored in Database.
- ✔ Rules, which are enforced on data being entered and prevents user from entering invalid data into tables are called “Constraints”.
- ✔ This ensures the accuracy and reliability of the data in the database.
- ✔ Constraints could be either on a column level or a table level.
- ✔ The column level constraints are applied only to one column, whereas the table level constraints are applied to the whole table.

Types of Integrity Constraints

1. Domain Integrity Constraints
2. Entity Integrity Constraints
3. Key Constraints
4. Referential Integrity Constraints



Integrity Constraints

Domain Constraint

Entity Constraint

Referential Constraint

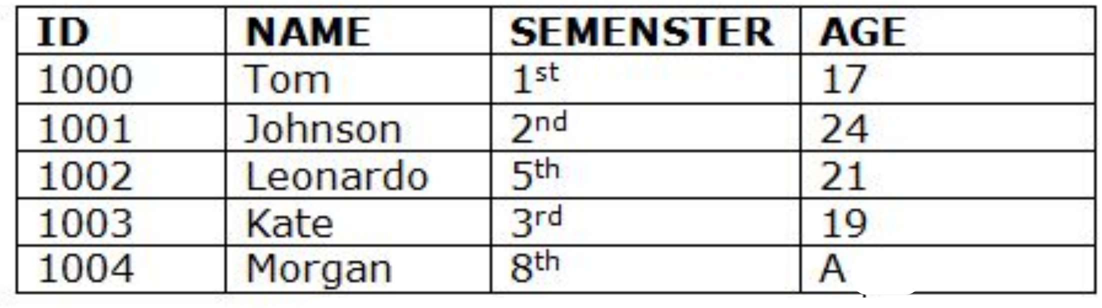
Key Constraint

1. <u>Domain Integrity Constraints</u>

- Domain constraints can be defined as the definition of a valid set of values for an attribute.
    - The data type of domain includes string, character, integer, time, date, currency, etc.
    - The value should fall in specified domain range
    - The value of the attribute must be available in the corresponding domain.
    - and so on….

1. Domain Integrity Constraints

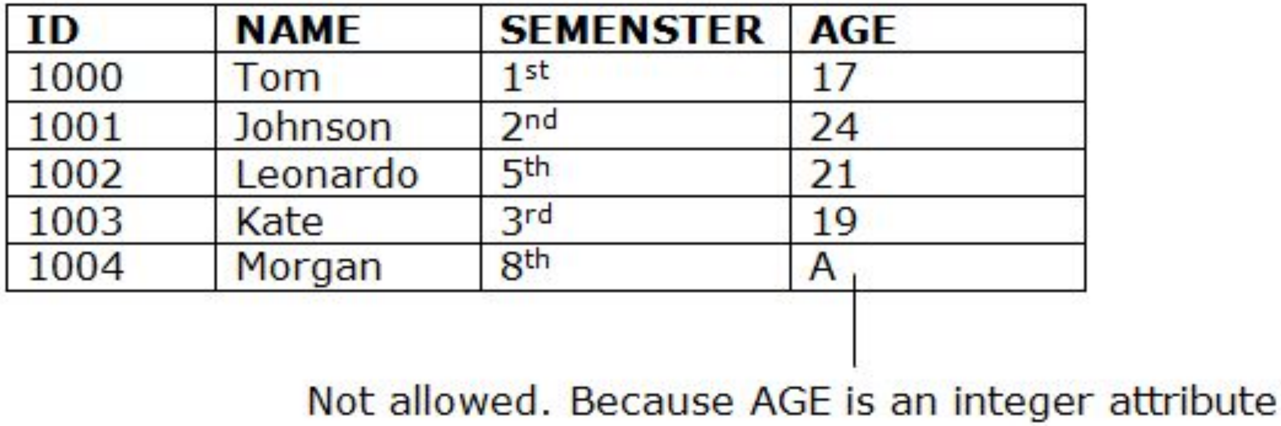
- Example



ID	NAME	SEMENSTER	AGE
1000	Tom	1st	17
1001	Johnson	2nd	24
1002	Leonardo	5th	21
1003	Kate	3rd	19
1004	Morgan	8th	A

1. **Domain Integrity Constraints**

- Example



ID	NAME	SEMENSTER	AGE
1000	Tom	1st	17
1001	Johnson	2nd	24
1002	Leonardo	5th	21
1003	Kate	3rd	19
1004	Morgan	8th	A

Not allowed. Because AGE is an integer attribute

1. Domain Integrity Constraints

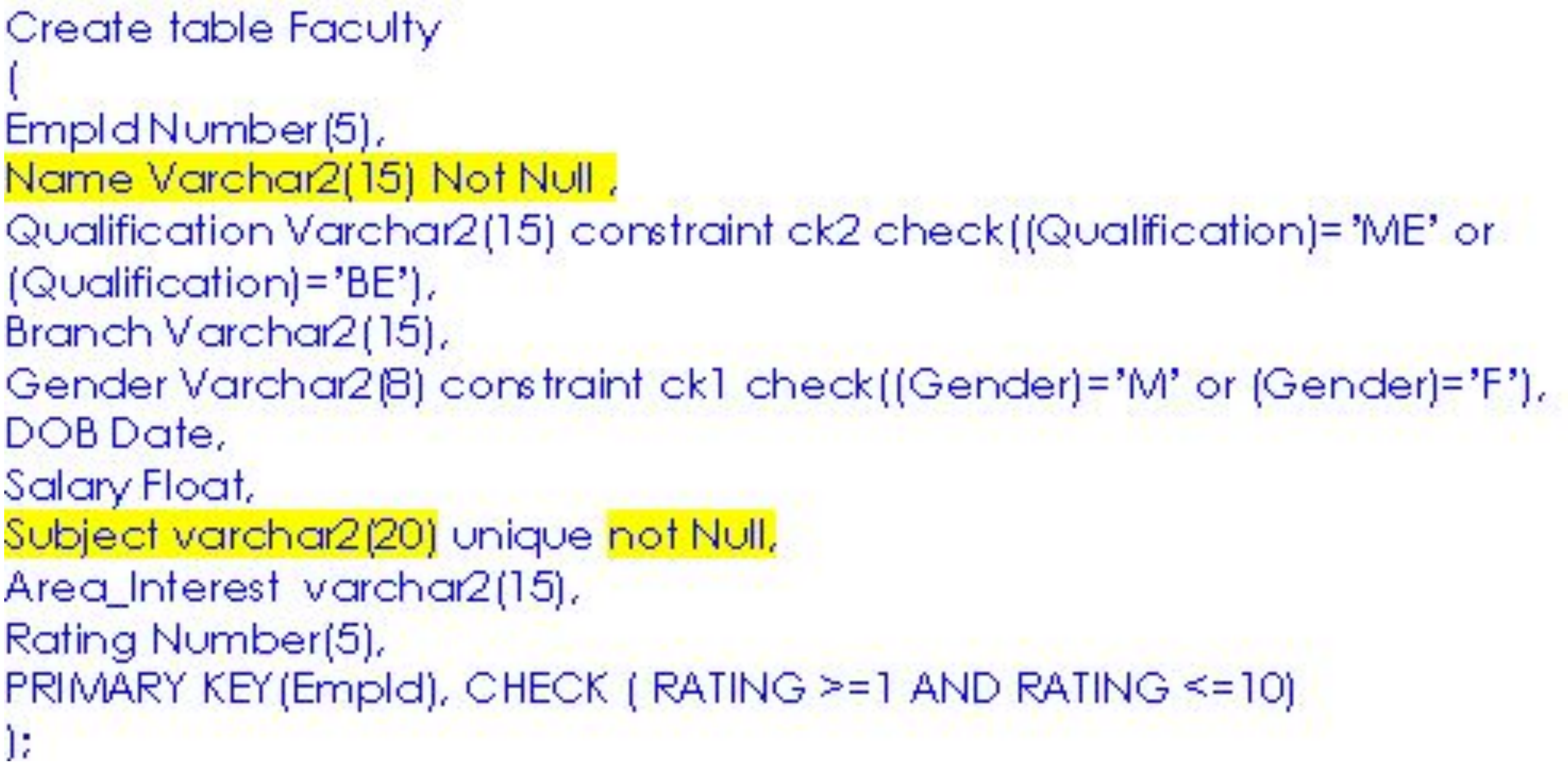
A. **Not Null Constraint – The enforcement of Not Null Constraints in a table ensures that the table contains values.**

For example: Each employee in college should assign one subject. That’s why it can not be null. So, enforces Employee Name & Subject fields to always contain a value.

1. Domain Integrity Constraints

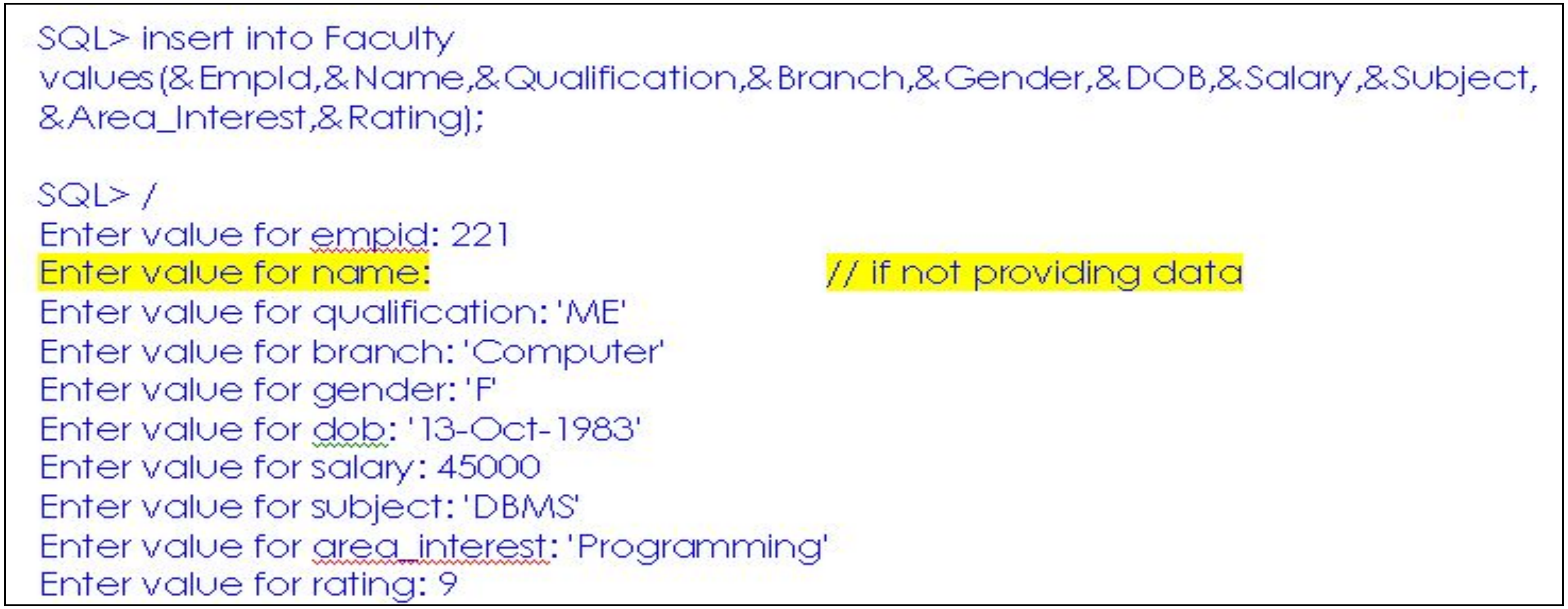
A. Not Null Constraint – The enforcement of Not Null Constraints in a table ensures that the table contains values.

For example: Each employee in college should assign one subject. That's why it can not be null. So, enforces Employee Name & Subject fields to always contain a value.



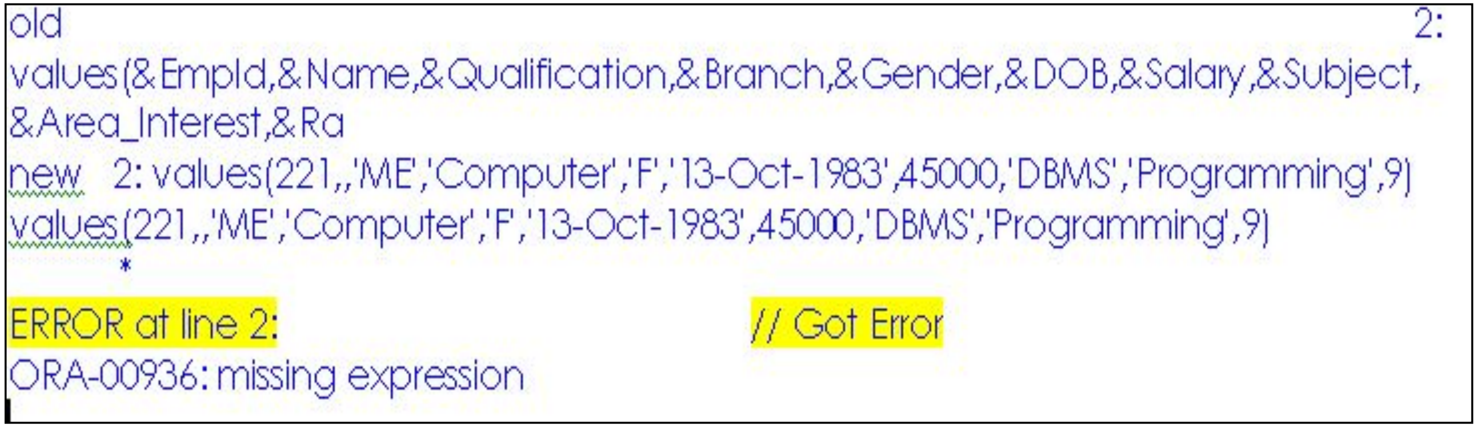
```
Create table Faculty
(
EmpId Number(5),
Name Varchar2(15) Not Null ,
Qualification Varchar2(15) constraint ck2 check((Qualification)='ME' or
(Qualification)='BE'),
Branch Varchar2(15),
Gender Varchar2(8) constraint ck1 check((Gender)='M' or (Gender)='F'),
DOB Date,
Salary Float,
Subject varchar2(20) unique not Null,
Area_Interest  varchar2(15),
Rating Number(5),
PRIMARY KEY(EmpId), CHECK ( RATING >=1 AND RATING <=10)
);
```

INPUT -- > if not providing data



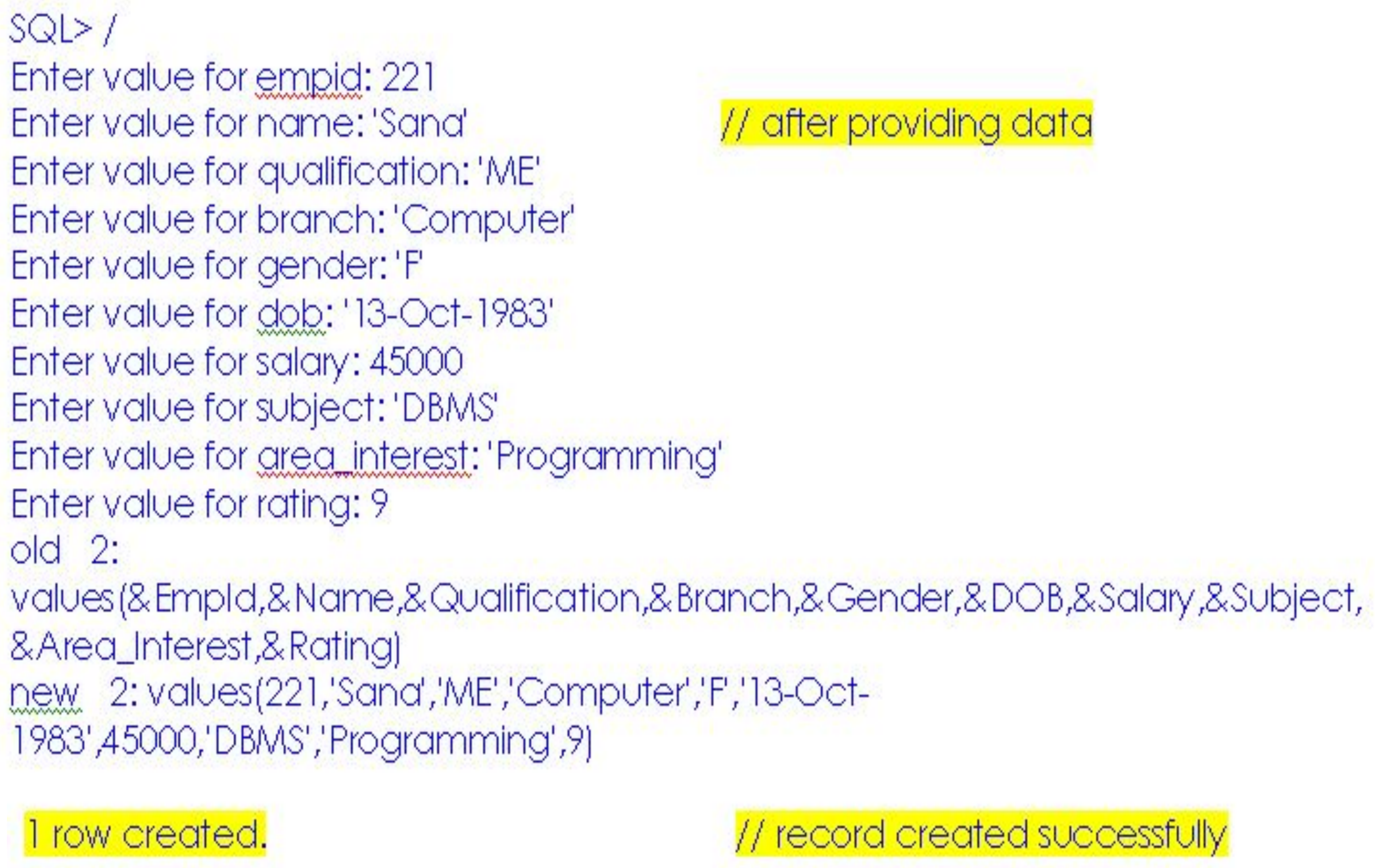
```
SQL> insert into Faculty
values(&EmpId,&Name,&Qualification,&Branch,&Gender,&DOB,&Salary,&Subject,
&Area_Interest,&Rating);

SQL> /
Enter value for empid: 221
Enter value for name:                          // if not providing data
Enter value for qualification: 'ME'
Enter value for branch: 'Computer'
Enter value for gender: 'F'
Enter value for dob: '13-Oct-1983'
Enter value for salary: 45000
Enter value for subject: 'DBMS'
Enter value for area_interest: 'Programming'
Enter value for rating: 9
```

OUTPUT



```
old                                                                  2:
values(&EmpId,&Name,&Qualification,&Branch,&Gender,&DOB,&Salary,&Subject,
&Area_Interest,&Ra
new   2: values(221,,'ME','Computer','F','13-Oct-1983',45000,'DBMS','Programming',9)
values(221,,'ME','Computer','F','13-Oct-1983',45000,'DBMS','Programming',9)
           *
ERROR at line 2:                          // Got Error
ORA-00936: missing expression
```

INPUT -- > After providing data



```
SQL> /
Enter value for empid: 221
Enter value for name: 'Sana'                         // after providing data
Enter value for qualification: 'ME'
Enter value for branch: 'Computer'
Enter value for gender: 'F'
Enter value for dob: '13-Oct-1983'
Enter value for salary: 45000
Enter value for subject: 'DBMS'
Enter value for area_interest: 'Programming'
Enter value for rating: 9
old   2:
values(&EmpId,&Name,&Qualification,&Branch,&Gender,&DOB,&Salary,&Subject,
&Area_Interest,&Rating)
new   2: values(221,'Sana','ME','Computer','F','13-Oct-
1983',45000,'DBMS','Programming',9)

 1 row created.                                      // record created successfully
```

OUTPUT

B. Check Constraint – Allow only a particular values or range of values

For example:

Create a table with following check constraints.

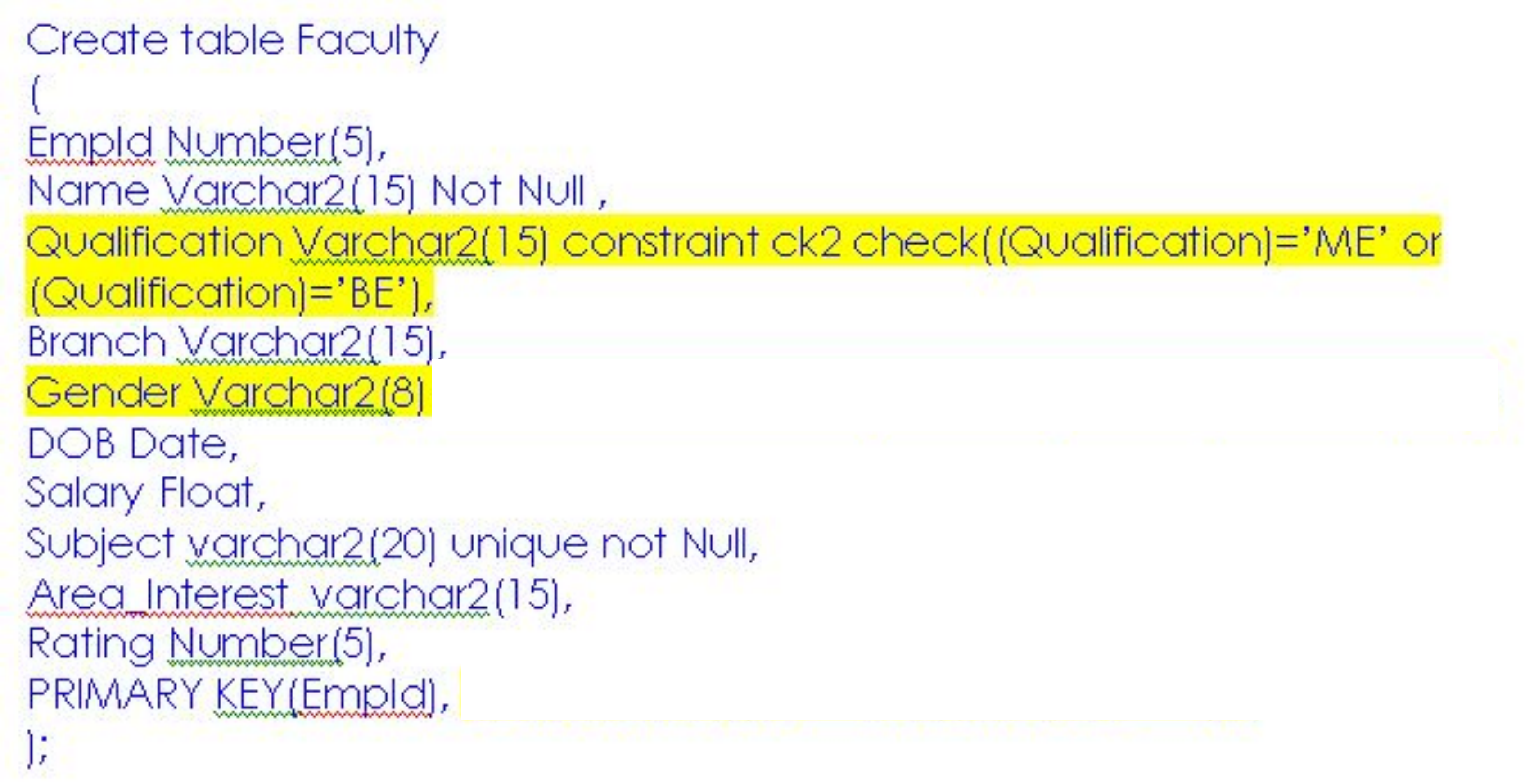
- **Only allow ‘BE’ or ‘ME’ values for the Qualification column.**
- **Only allow ‘F’ or ‘M’ values for the Gender column.**
- **Rate each employee in the range 1 to 10**

B. Check Constraint – Allow only a particular values or range of values

For example:

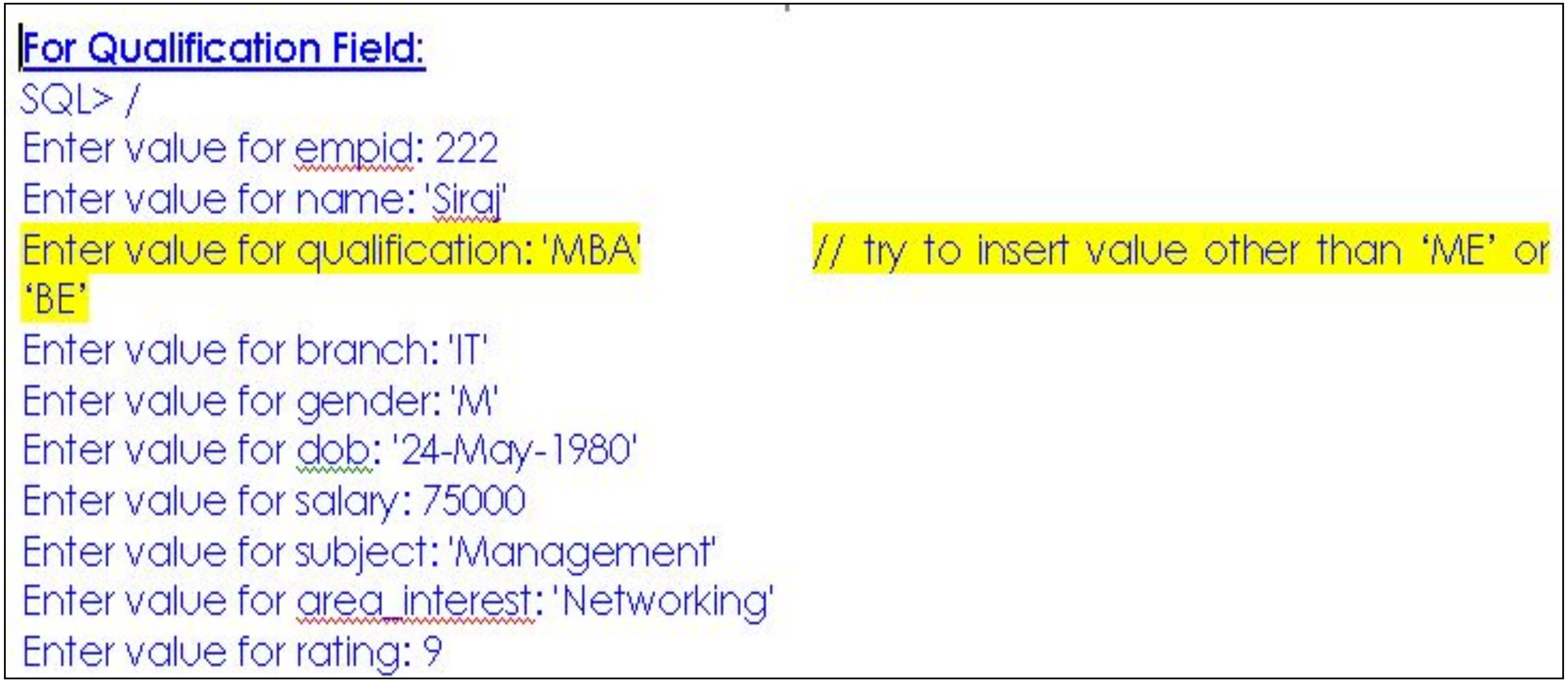
Create a table with following check constraints.

- **Only allow ‘BE’ or ‘ME’ values for the Qualification column.**
- **Only allow ‘F’ or ‘M’ values for the Gender column.**
- **Rate each employee in the range 1 to 10**



```
Create table Faculty
(
EmpId Number(5),
Name Varchar2(15) Not Null ,
Qualification Varchar2(15) constraint ck2 check((Qualification)='ME' or
(Qualification)='BE'),
Branch Varchar2(15),
Gender Varchar2(8)
DOB Date,
Salary Float,
Subject varchar2(20) unique not Null,
Area_Interest  varchar2(15),
Rating Number(5),
PRIMARY KEY(EmpId),
);
```

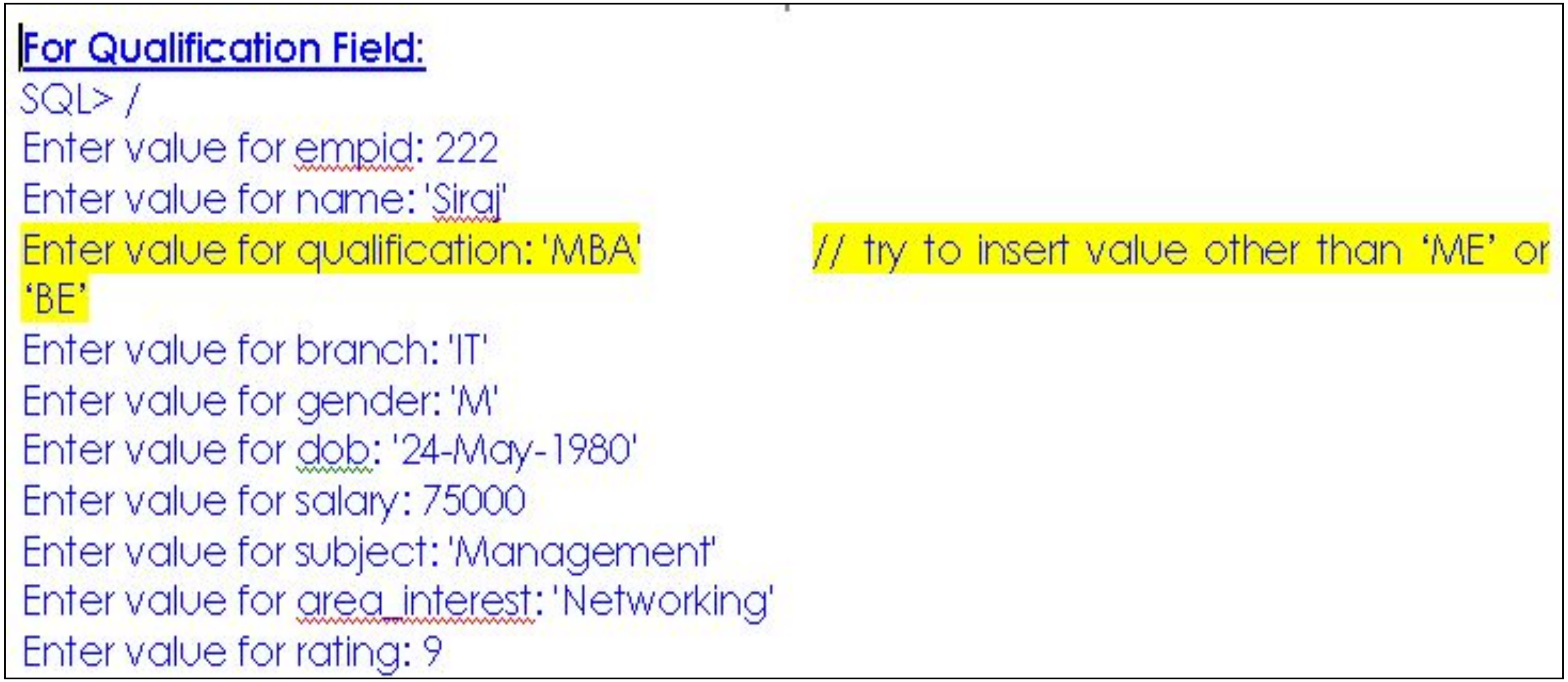
INPUT -- > try to insert value other than ‘ME’ or ‘BE’ for Qualification column



For Qualification Field:

```
SQL> /
Enter value for empid: 222
Enter value for name: 'Siraj'
Enter value for qualification: 'MBA'          // try to insert value other than ‘ME’ or
‘BE’
Enter value for branch: 'IT'
Enter value for gender: 'M'
Enter value for dob: '24-May-1980'
Enter value for salary: 75000
Enter value for subject: 'Management'
Enter value for area_interest: 'Networking'
Enter value for rating: 9
```

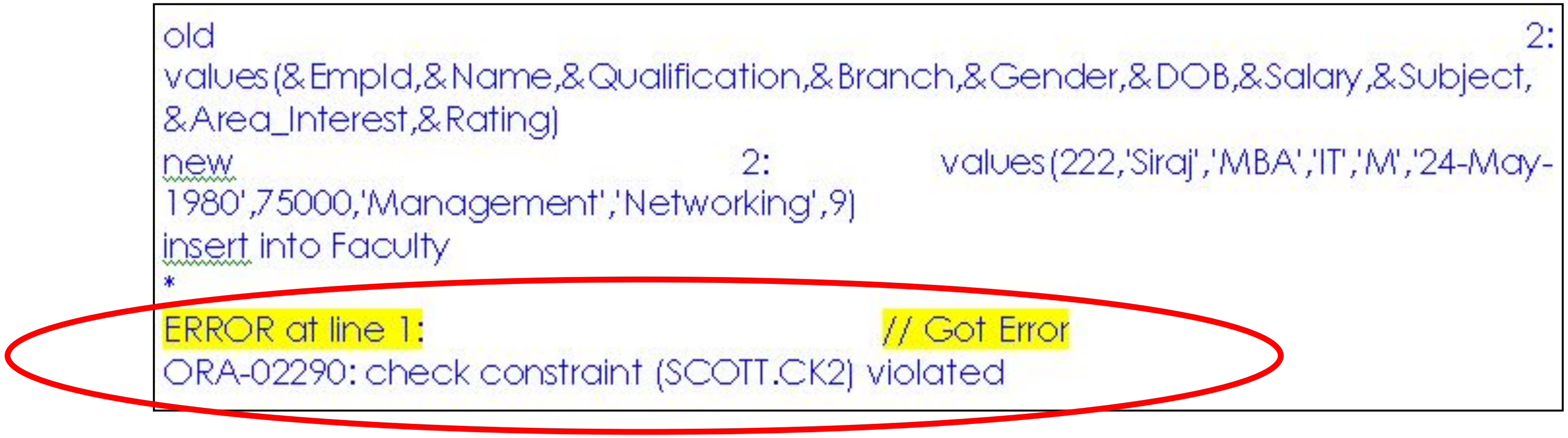
INPUT -- > try to insert value other than ‘ME’ or ‘BE’ for Qualification column



For Qualification Field:

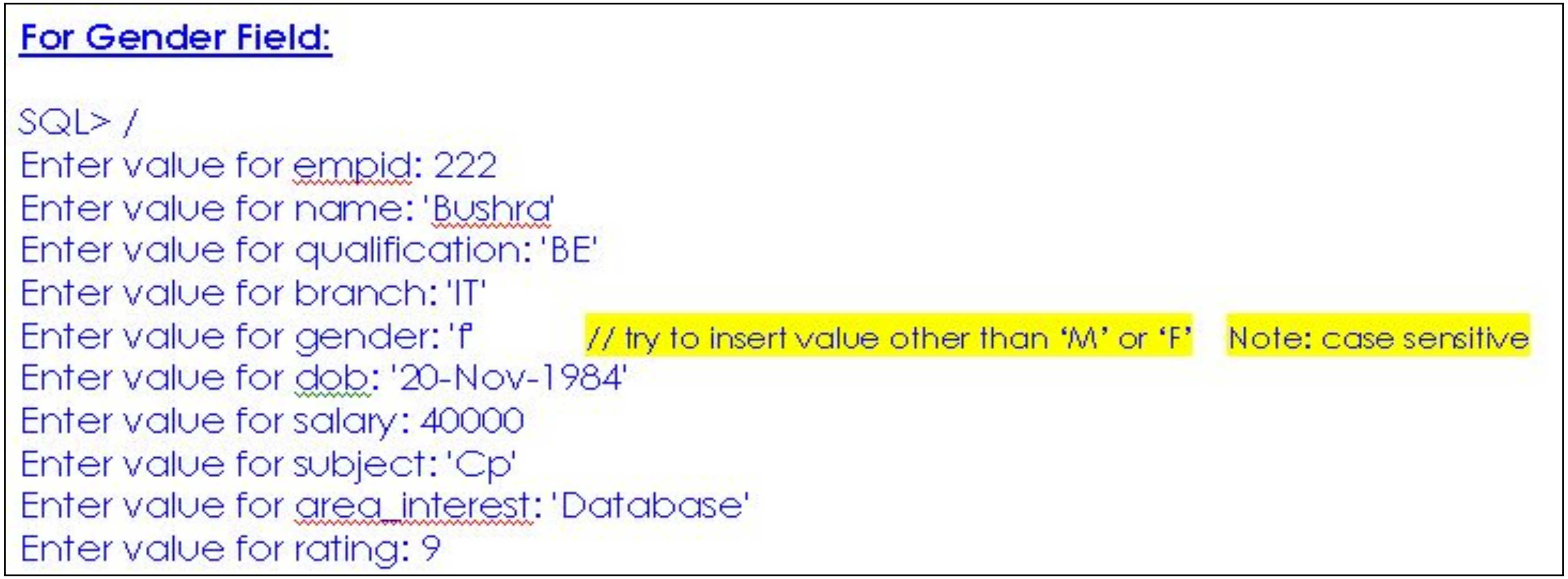
```
SQL> /
Enter value for empid: 222
Enter value for name: 'Siraj'
Enter value for qualification: 'MBA'          // try to insert value other than ‘ME’ or
‘BE’
Enter value for branch: 'IT'
Enter value for gender: 'M'
Enter value for dob: '24-May-1980'
Enter value for salary: 75000
Enter value for subject: 'Management'
Enter value for area_interest: 'Networking'
Enter value for rating: 9
```

OUTPUT



```
old                                                                    2:
values(&EmpId,&Name,&Qualification,&Branch,&Gender,&DOB,&Salary,&Subject,
&Area_Interest,&Rating)
new                      2:          values(222,'Siraj','MBA','IT','M','24-May-
1980',75000,'Management','Networking',9)
insert into Faculty
*
ERROR at line 1:                           // Got Error
ORA-02290: check constraint (SCOTT.CK2) violated
```

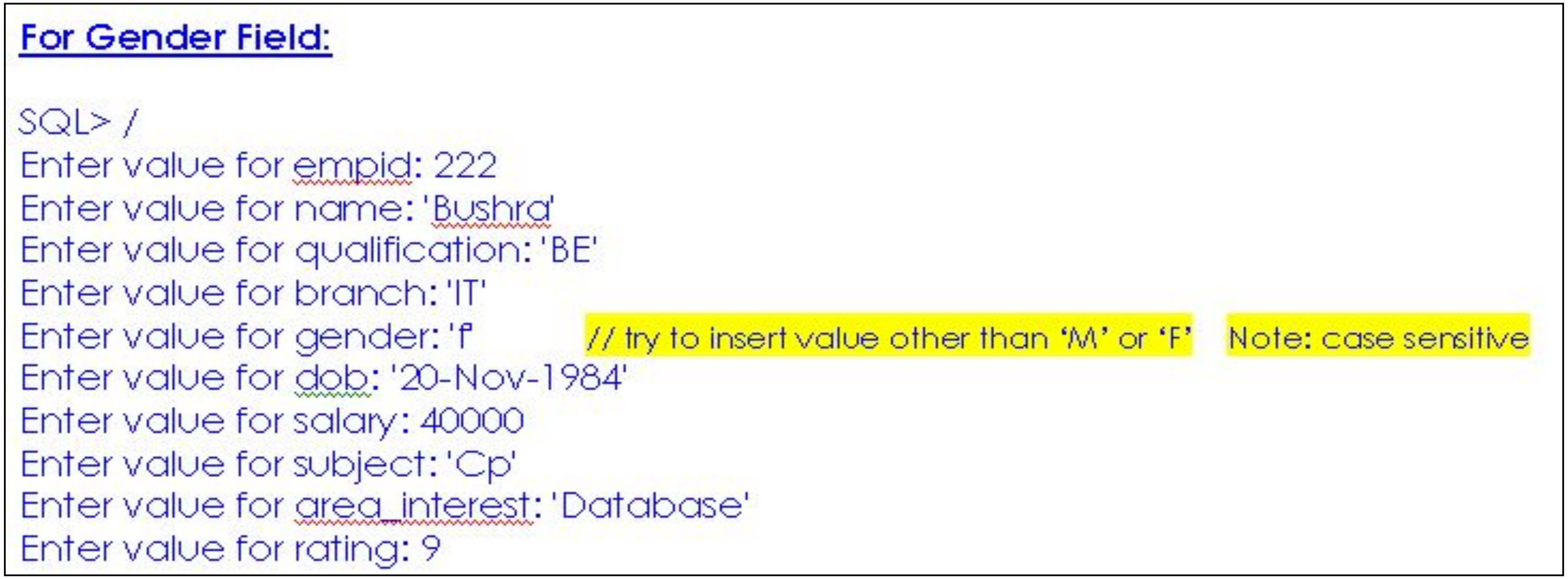
INPUT -- > try to insert value other than ‘M’ or ‘F’



For Gender Field:

```
SQL> /
Enter value for empid: 222
Enter value for name: 'Bushra'
Enter value for qualification: 'BE'
Enter value for branch: 'IT'
Enter value for gender: 'f'          // try to insert value other than ‘M’ or ‘F’    Note: case sensitive
Enter value for dob: '20-Nov-1984'
Enter value for salary: 40000
Enter value for subject: 'Cp'
Enter value for area_interest: 'Database'
Enter value for rating: 9
```

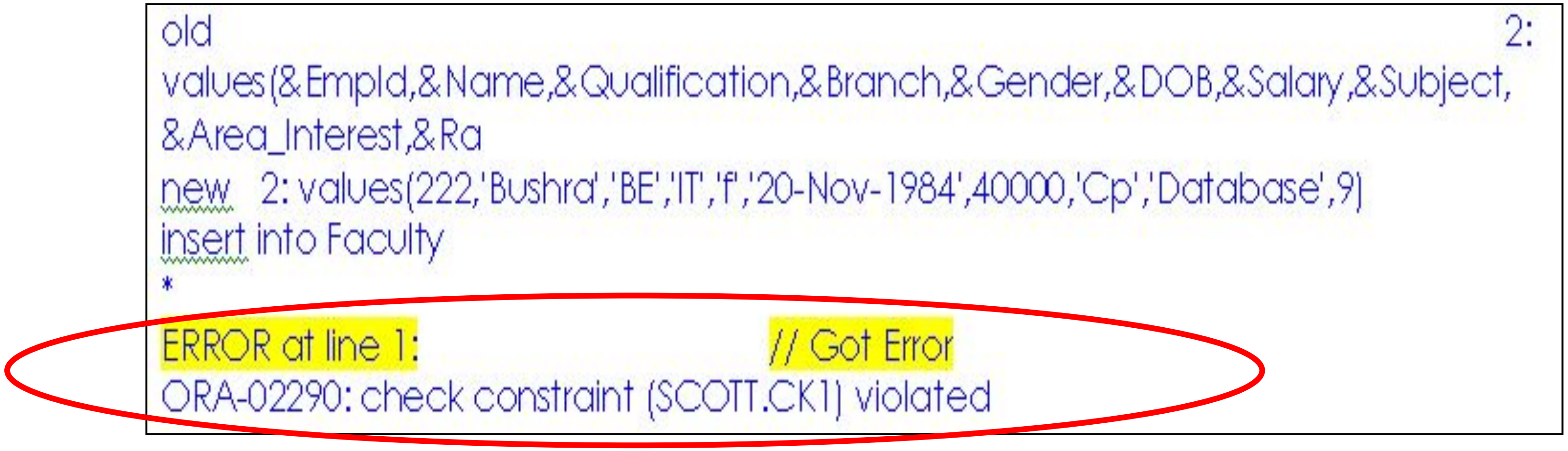
INPUT -- > try to insert value other than ‘M’ or ‘F’



For Gender Field:

```
SQL> /
Enter value for empid: 222
Enter value for name: 'Bushra'
Enter value for qualification: 'BE'
Enter value for branch: 'IT'
Enter value for gender: 'f'          // try to insert value other than ‘M’ or ‘F’   Note: case sensitive
Enter value for dob: '20-Nov-1984'
Enter value for salary: 40000
Enter value for subject: 'Cp'
Enter value for area_interest: 'Database'
Enter value for rating: 9
```

OUTPUT



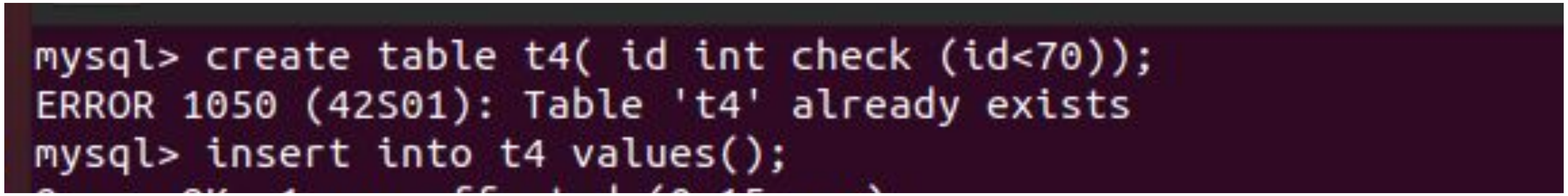
```
old                                                                   2:
values(&EmpId,&Name,&Qualification,&Branch,&Gender,&DOB,&Salary,&Subject,
&Area_Interest,&Ra
new   2: values(222,'Bushra','BE','IT','f','20-Nov-1984',40000,'Cp','Database',9)
insert into Faculty
*
ERROR at line 1:                   // Got Error
ORA-02290: check constraint (SCOTT.CK1) violated
```

Example:

```
mysql> create table t4 ( id int check (id<70));
```

Example:

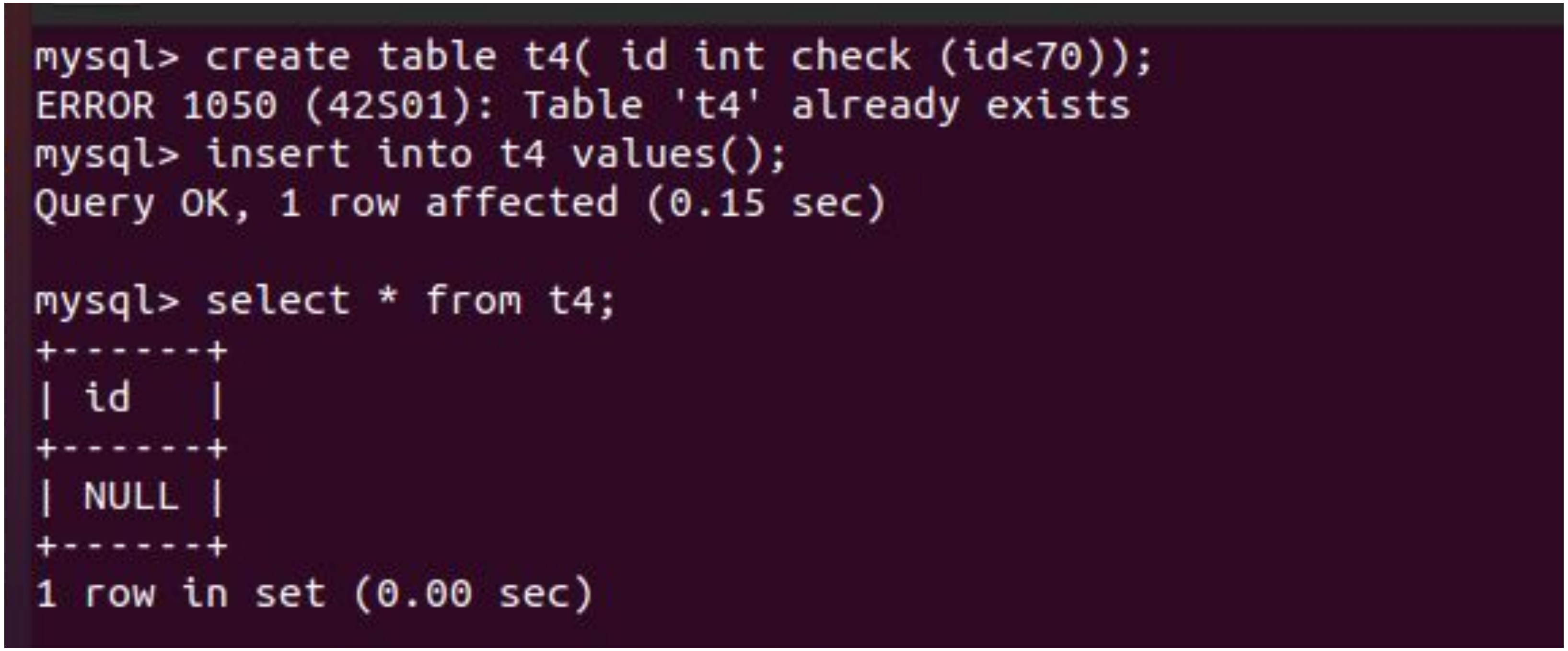
mysql> create table t4 (id int check (id<70));



```
mysql> create table t4( id int check (id<70));
ERROR 1050 (42S01): Table 't4' already exists
mysql> insert into t4 values();
```

Example:

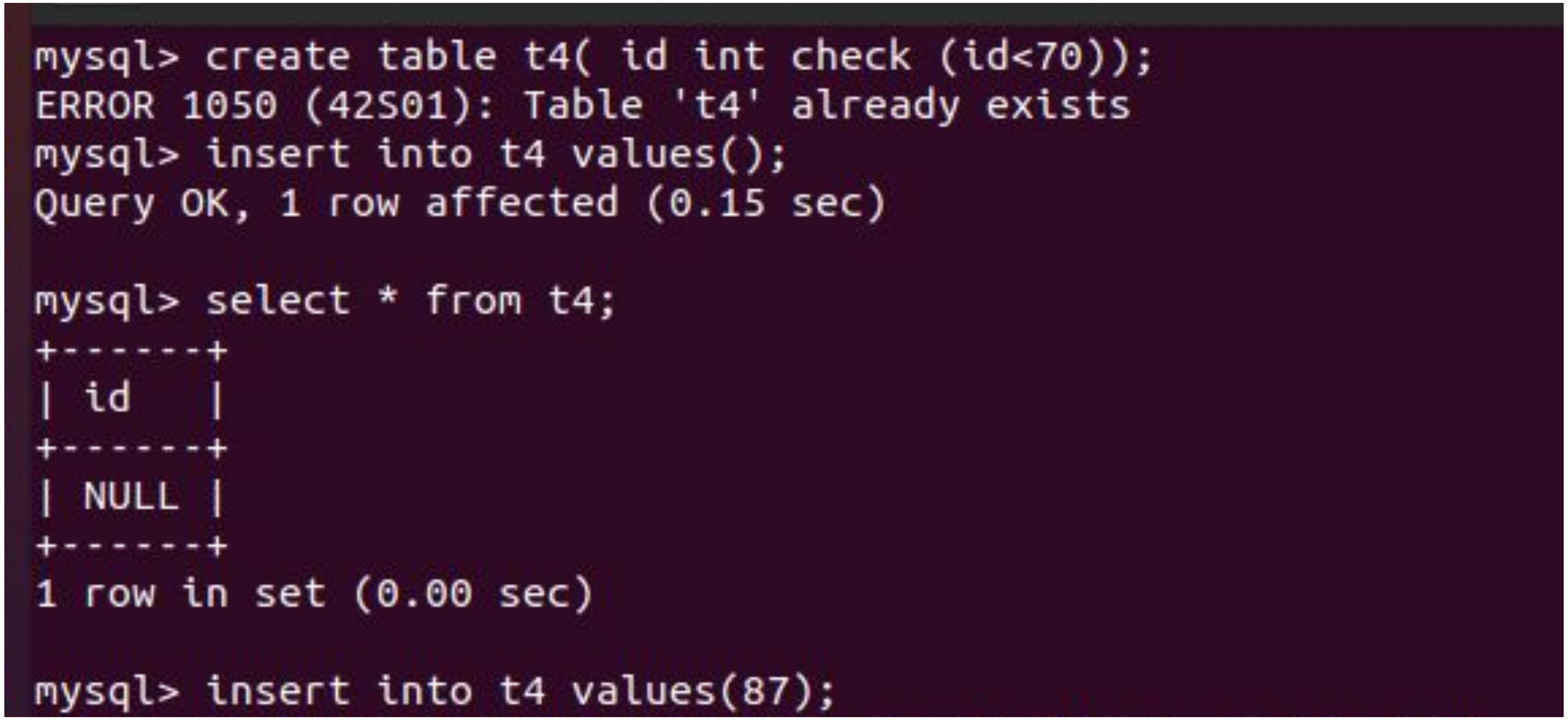
mysql> create table t4 (id int check (id<70));



```
mysql> create table t4( id int check (id<70));
ERROR 1050 (42S01): Table 't4' already exists
mysql> insert into t4 values();
Query OK, 1 row affected (0.15 sec)

mysql> select * from t4;
+------+
| id   |
+------+
| NULL |
+------+
1 row in set (0.00 sec)
```

Example:

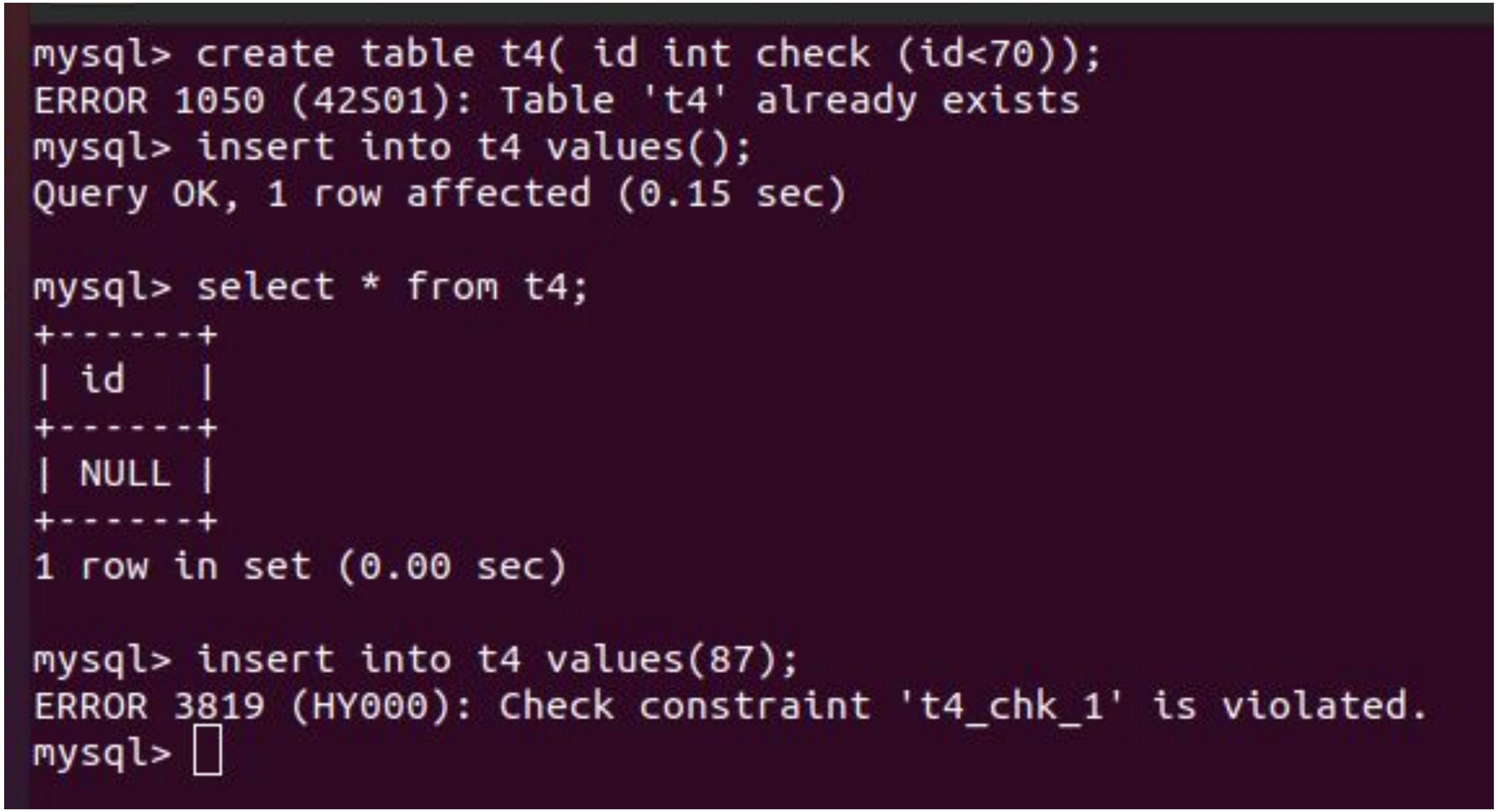
mysql> create table t4 (id int check (id<70));



```
mysql> create table t4( id int check (id<70));
ERROR 1050 (42S01): Table 't4' already exists
mysql> insert into t4 values();
Query OK, 1 row affected (0.15 sec)

mysql> select * from t4;
+------+
| id   |
+------+
| NULL |
+------+
1 row in set (0.00 sec)

mysql> insert into t4 values(87);
```

Example:

mysql> create table t4 (id int check (id<70));



```
mysql> create table t4( id int check (id<70));
ERROR 1050 (42S01): Table 't4' already exists
mysql> insert into t4 values();
Query OK, 1 row affected (0.15 sec)

mysql> select * from t4;
+------+
| id   |
+------+
| NULL |
+------+
1 row in set (0.00 sec)

mysql> insert into t4 values(87);
ERROR 3819 (HY000): Check constraint 't4_chk_1' is violated.
mysql> 
```

2. Entity Integrity Constraints

- **To identify each row in a table, the table must have a primary key.**

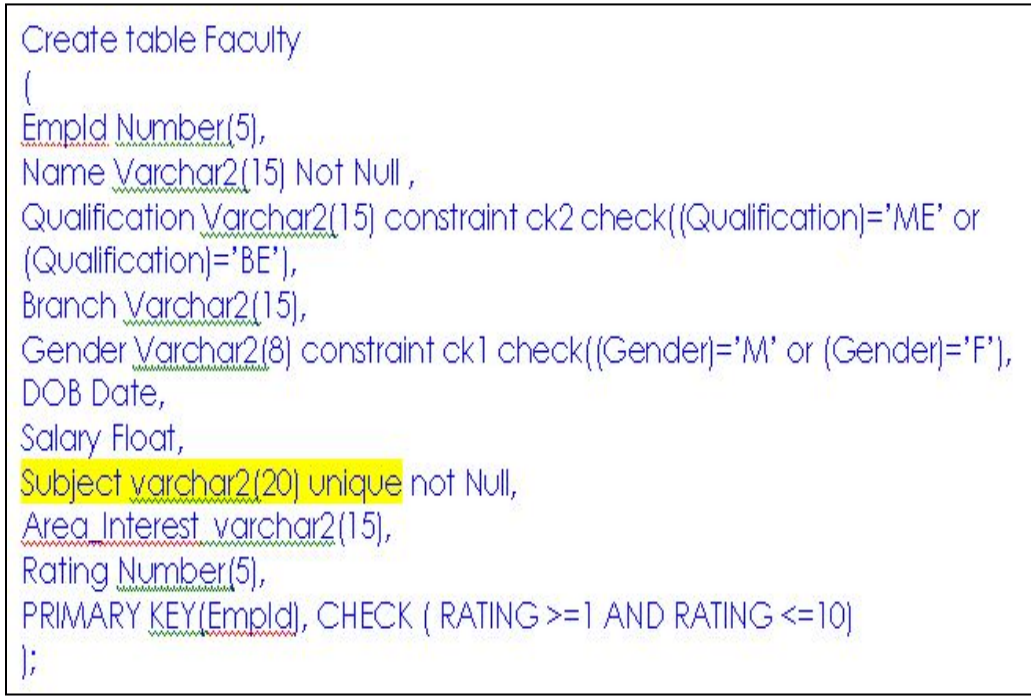
- **The primary key is a unique value that identifies each row. This requirement is called the *entity integrity constraint*.**

A. Unique Constraints

- ✔ The unique constraint designates a Column or a group of columns as Unique key.
- ✔ This allows only unique value to be stored in the column.
- ✔ Rejects duplication.
- ✔ *Therefore unique key constraints ensure that information in the column(s) must not be repeated.*

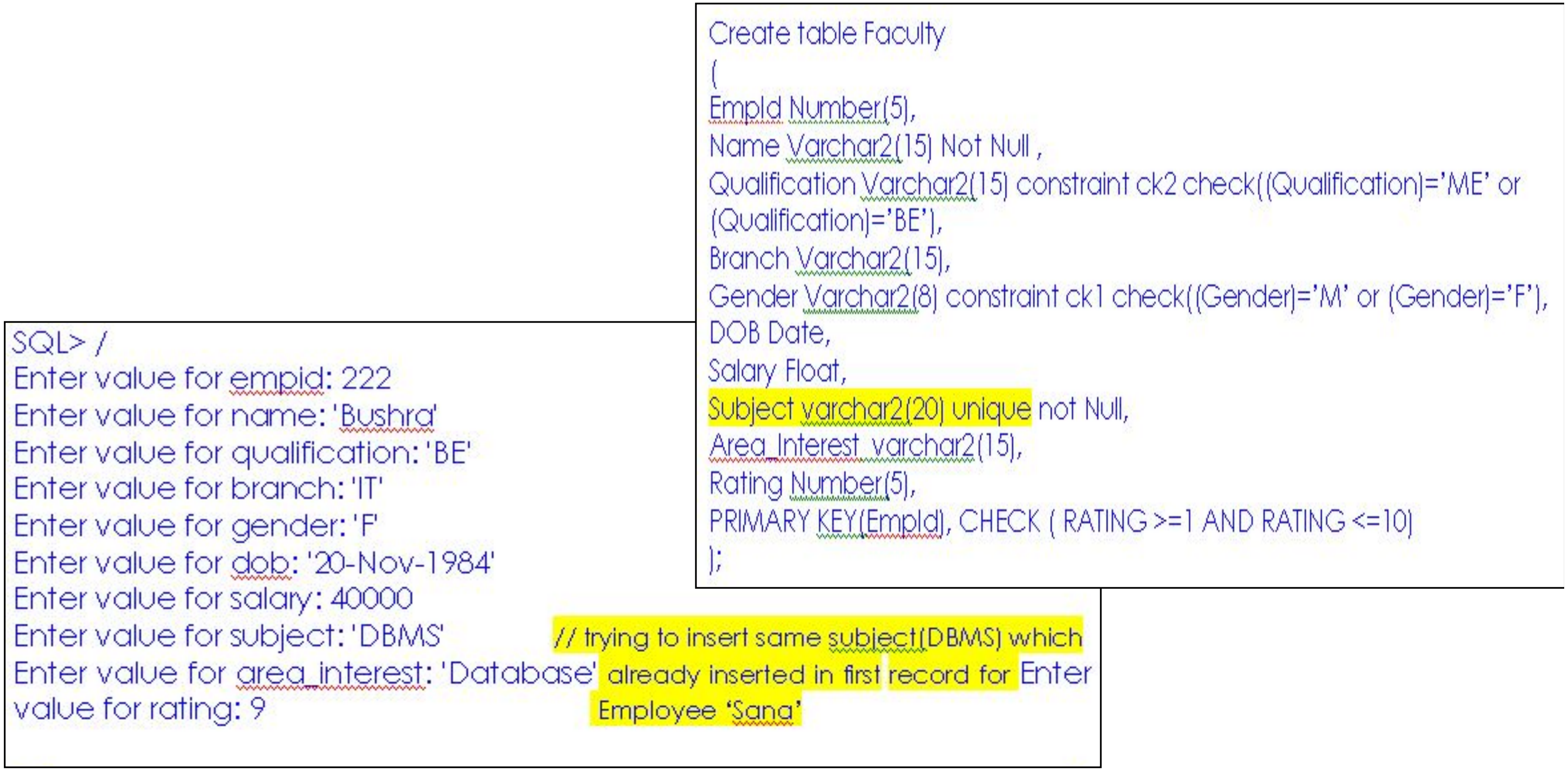
For example each Teacher has assigned different subjects.

For example each Teacher has assigned different subjects.



```
Create table Faculty
(
EmpId Number(5),
Name Varchar2(15) Not Null ,
Qualification Varchar2(15) constraint ck2 check((Qualification)='ME' or
(Qualification)='BE'),
Branch Varchar2(15),
Gender Varchar2(8) constraint ck1 check((Gender)='M' or (Gender)='F'),
DOB Date,
Salary Float,
Subject varchar2(20) unique not Null,
Area_Interest_varchar2(15),
Rating Number(5),
PRIMARY KEY(EmpId), CHECK ( RATING >=1 AND RATING <=10)
);
```

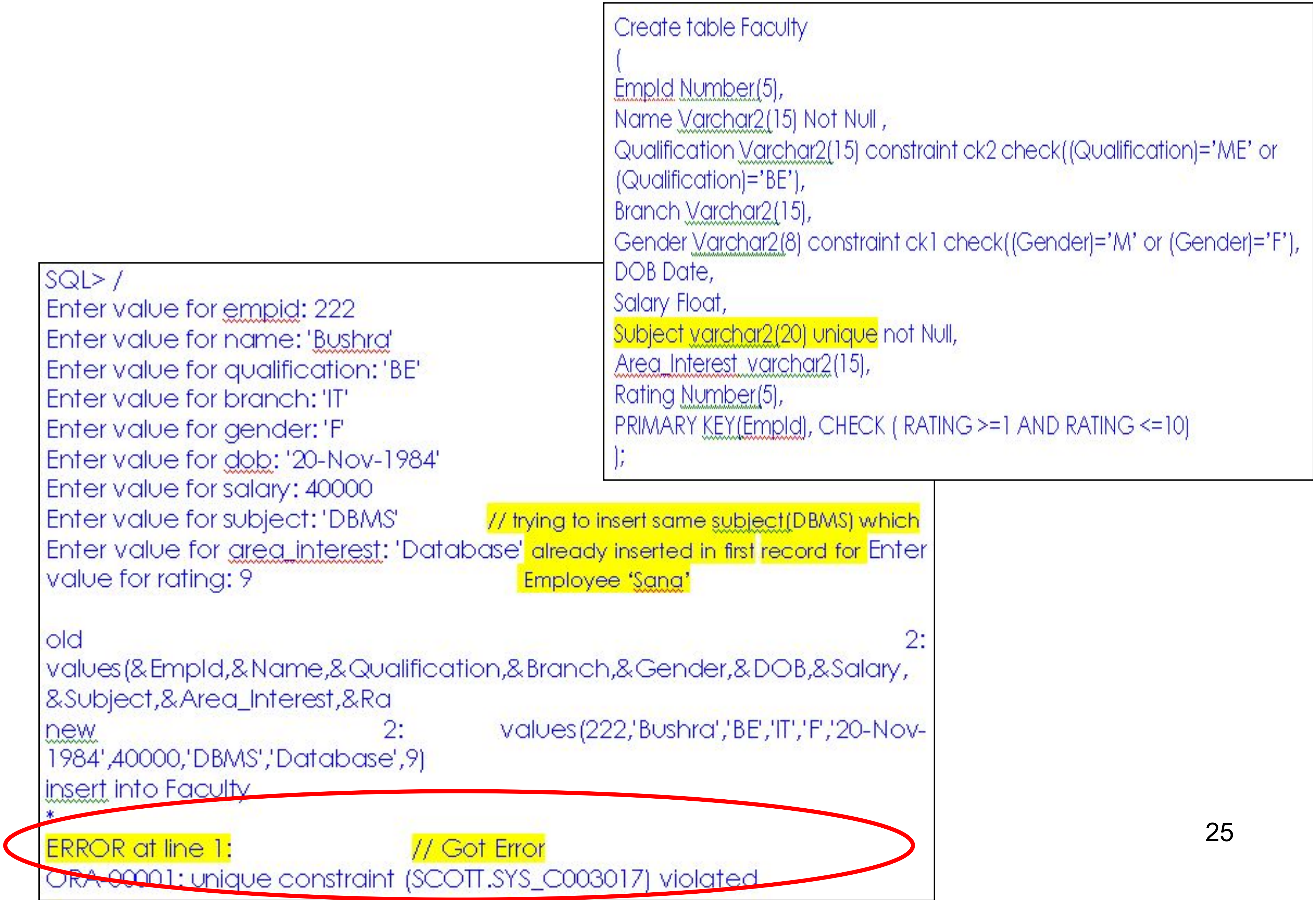
For example each Teacher has assigned different subjects.



```
Create table Faculty
(
EmpId Number(5),
Name Varchar2(15) Not Null ,
Qualification Varchar2(15) constraint ck2 check((Qualification)='ME' or
(Qualification)='BE'),
Branch Varchar2(15),
Gender Varchar2(8) constraint ck1 check((Gender)='M' or (Gender)='F'),
DOB Date,
Salary Float,
Subject varchar2(20) unique not Null,
Area_Interest varchar2(15),
Rating Number(5),
PRIMARY KEY(EmpId), CHECK ( RATING >=1 AND RATING <=10)
);
```

```
SQL> /
Enter value for empid: 222
Enter value for name: 'Bushra'
Enter value for qualification: 'BE'
Enter value for branch: 'IT'
Enter value for gender: 'F'
Enter value for dob: '20-Nov-1984'
Enter value for salary: 40000
Enter value for subject: 'DBMS'             // trying to insert same subject(DBMS) which
Enter value for area_interest: 'Database'  already inserted in first record for Enter
value for rating: 9                           Employee 'Sana'
```

For example each Teacher has assigned different subjects.



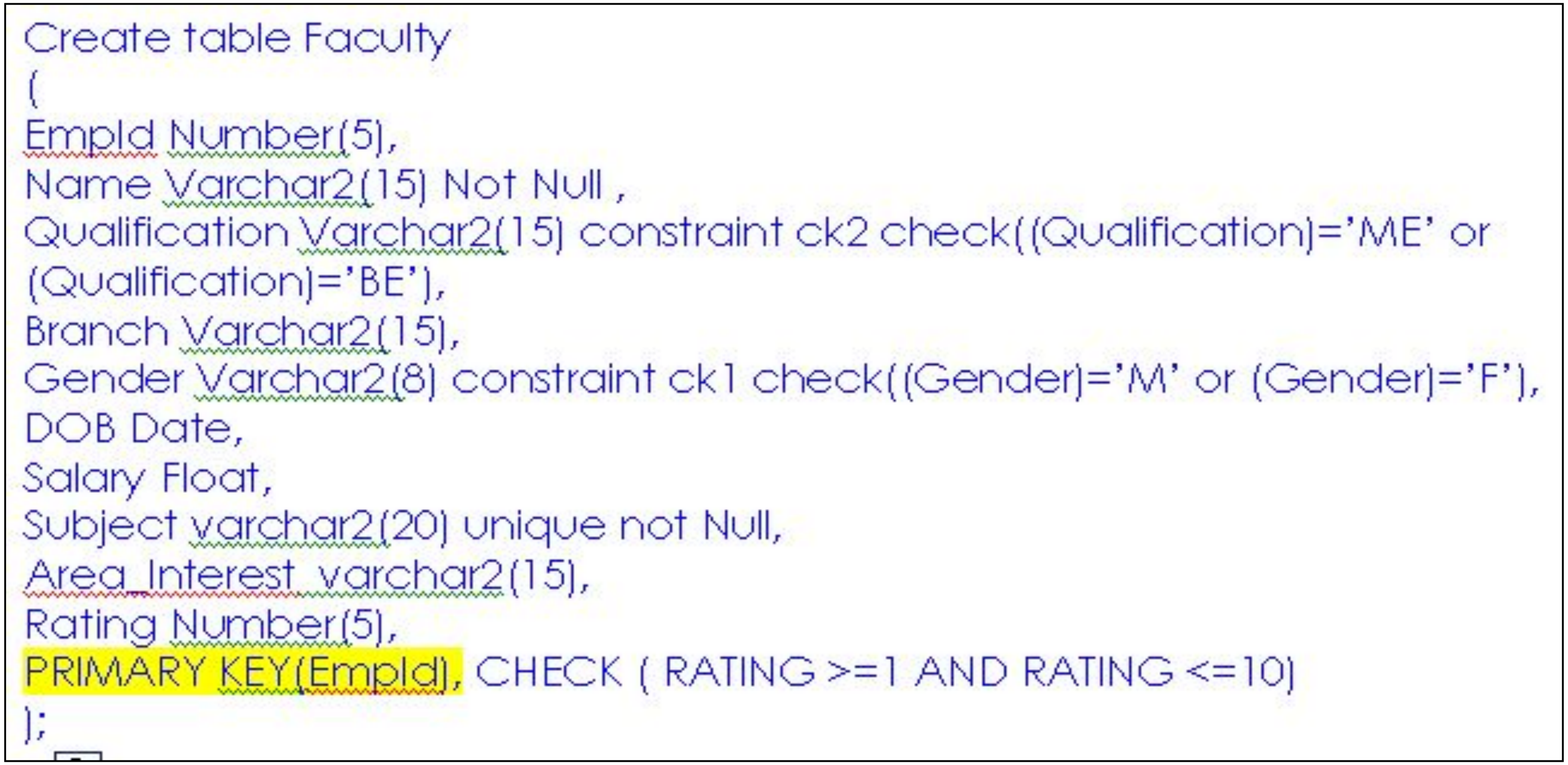
```
Create table Faculty
(
EmpId Number(5),
Name Varchar2(15) Not Null ,
Qualification Varchar2(15) constraint ck2 check((Qualification)=’ME’ or
(Qualification)=’BE’),
Branch Varchar2(15),
Gender Varchar2(8) constraint ck1 check((Gender)=’M’ or (Gender)=’F’),
DOB Date,
Salary Float,
Subject varchar2(20) unique not Null,
Area_Interest varchar2(15),
Rating Number(5),
PRIMARY KEY(EmpId), CHECK ( RATING >=1 AND RATING <=10)
);
```

```
SQL> /
Enter value for empid: 222
Enter value for name: 'Bushra'
Enter value for qualification: 'BE'
Enter value for branch: 'IT'
Enter value for gender: 'F'
Enter value for dob: '20-Nov-1984'
Enter value for salary: 40000
Enter value for subject: 'DBMS'          // trying to insert same subject(DBMS) which
Enter value for area_interest: 'Database' already inserted in first record for Enter
value for rating: 9                         Employee ‘Sana’

old                                                                  2:
values(&EmpId,&Name,&Qualification,&Branch,&Gender,&DOB,&Salary,
&Subject,&Area_Interest,&Ra
new                          2:          values(222,'Bushra','BE','IT','F','20-Nov-
1984',40000,'DBMS','Database',9)
insert into Faculty
*
ERROR at line 1:                      // Got Error
ORA-00001: unique constraint (SCOTT.SYS_C003017) violated
```

B. Primary Key Constraints

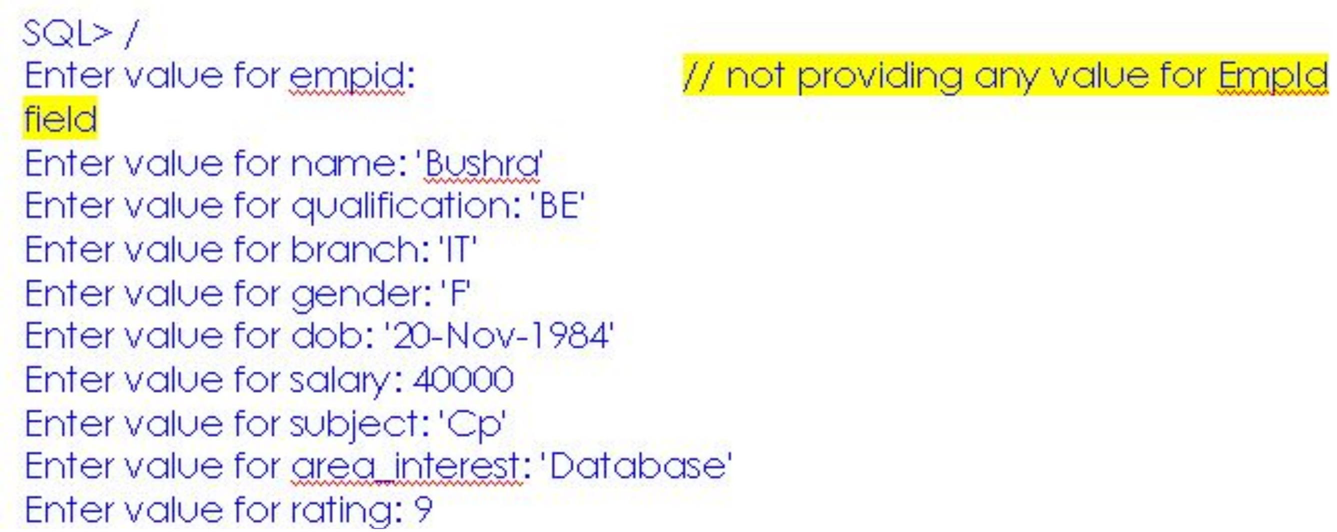
Primary key similar to unique key.

- Avoids duplication
- relation between two tables
- does not allow null values.



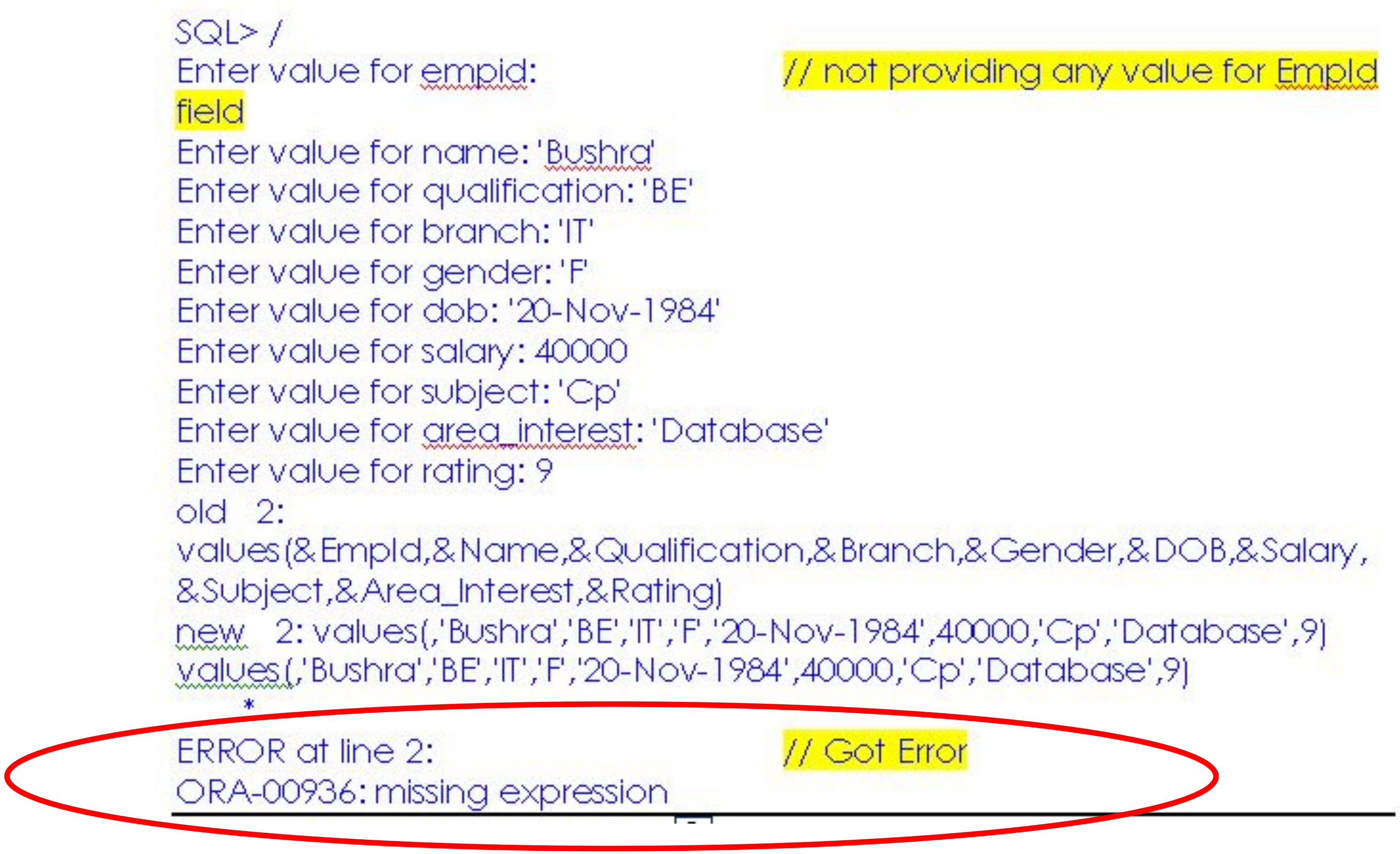
```
Create table Faculty
(
EmpId Number(5),
Name Varchar2(15) Not Null ,
Qualification Varchar2(15) constraint ck2 check((Qualification)='ME' or
(Qualification)='BE'),
Branch Varchar2(15),
Gender Varchar2(8) constraint ck1 check((Gender)='M' or (Gender)='F'),
DOB Date,
Salary Float,
Subject varchar2(20) unique not Null,
Area_Interest  varchar2(15),
Rating Number(5),
PRIMARY KEY(EmpId), CHECK ( RATING >=1 AND RATING <=10)
);
```

// If not providing any value for EmpId field (Primary key)



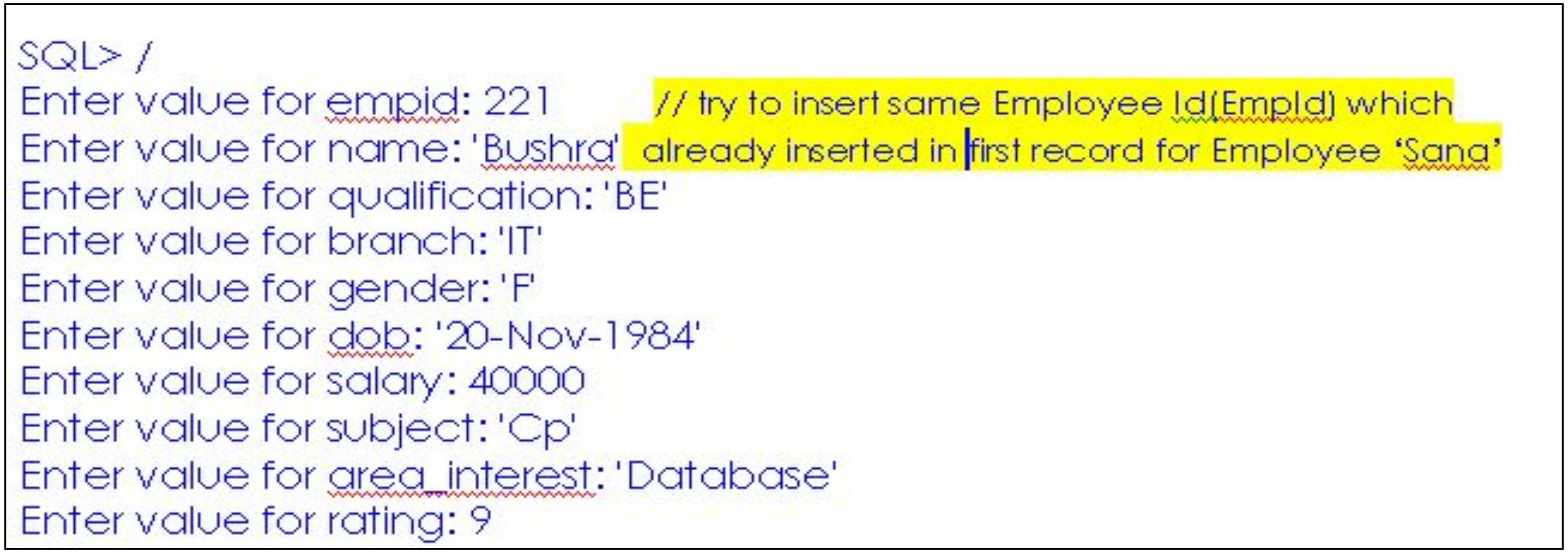
```
SQL> /
Enter value for empid:                    // not providing any value for EmpId
field
Enter value for name: 'Bushra'
Enter value for qualification: 'BE'
Enter value for branch: 'IT'
Enter value for gender: 'F'
Enter value for dob: '20-Nov-1984'
Enter value for salary: 40000
Enter value for subject: 'Cp'
Enter value for area_interest: 'Database'
Enter value for rating: 9
```

// If not providing any value for EmpId field (Primary key)



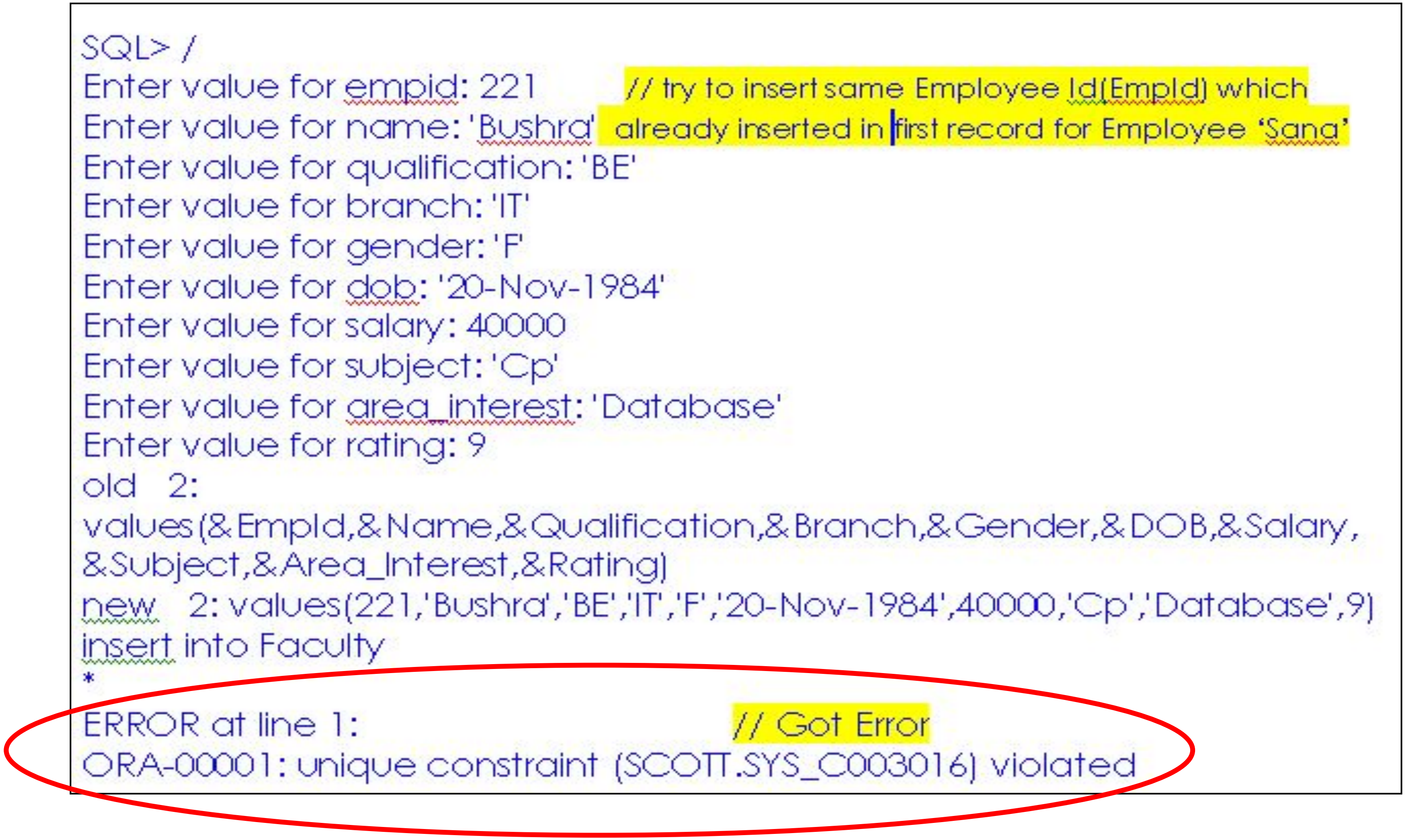
```
SQL> /
Enter value for empid:                    // not providing any value for EmpId
field
Enter value for name: 'Bushra'
Enter value for qualification: 'BE'
Enter value for branch: 'IT'
Enter value for gender: 'F'
Enter value for dob: '20-Nov-1984'
Enter value for salary: 40000
Enter value for subject: 'Cp'
Enter value for area_interest: 'Database'
Enter value for rating: 9
old   2:
values(&EmpId,&Name,&Qualification,&Branch,&Gender,&DOB,&Salary,
&Subject,&Area_Interest,&Rating)
new   2: values(,'Bushra','BE','IT','F','20-Nov-1984',40000,'Cp','Database',9)
values(,'Bushra','BE','IT','F','20-Nov-1984',40000,'Cp','Database',9)
      *
ERROR at line 2:                          // Got Error
ORA-00936: missing expression
```

// try to insert same Employee Id(EmpId) which already inserted in first record for Employee ‘Sana’ ’



```
SQL> /
Enter value for empid: 221        // try to insert same Employee Id(EmpId) which
Enter value for name: 'Bushra'  already inserted in first record for Employee ‘Sana’
Enter value for qualification: 'BE'
Enter value for branch: 'IT'
Enter value for gender: 'F'
Enter value for dob: '20-Nov-1984'
Enter value for salary: 40000
Enter value for subject: 'Cp'
Enter value for area_interest: 'Database'
Enter value for rating: 9
```

// try to insert same Employee Id(EmpId) which already inserted in first record for Employee ‘Sana’ ’



```
SQL> /
Enter value for empid: 221         // try to insert same Employee Id(EmpId) which
Enter value for name: 'Bushra'  already inserted in first record for Employee ‘Sana’
Enter value for qualification: 'BE'
Enter value for branch: 'IT'
Enter value for gender: 'F'
Enter value for dob: '20-Nov-1984'
Enter value for salary: 40000
Enter value for subject: 'Cp'
Enter value for area_interest: 'Database'
Enter value for rating: 9
old   2:
values(&EmpId,&Name,&Qualification,&Branch,&Gender,&DOB,&Salary,
&Subject,&Area_Interest,&Rating)
new   2: values(221,'Bushra','BE','IT','F','20-Nov-1984',40000,'Cp','Database',9)
insert into Faculty
*
ERROR at line 1:                                  // Got Error
ORA-00001: unique constraint (SCOTT.SYS_C003016) violated
```

3. Referential Integrity Constraints

Often we wish to ensure that a value appearing in a relation for a given set of attributes also appears for another set of attributes in another relation. This is called *referential integrity.*

Example:

Consider the Sailors relation and the constraint that no two sailors have the same SID. And, a sailor can’t reserve a boat unless he/she is a valid sailor.

3. Referential Integrity Constraints

Often we wish to ensure that a value appearing in a relation for a given set of attributes also appears for another set of attributes in another relation. This is called *referential integrity.*

Example:

Consider the Sailors relation and the constraint that no two sailors have the same SID. And, a sailor can't reserve a boat unless he/she is a valid sailor.

Sailor Table:

Sid
Sname
Rating
Age

Reserve Table:

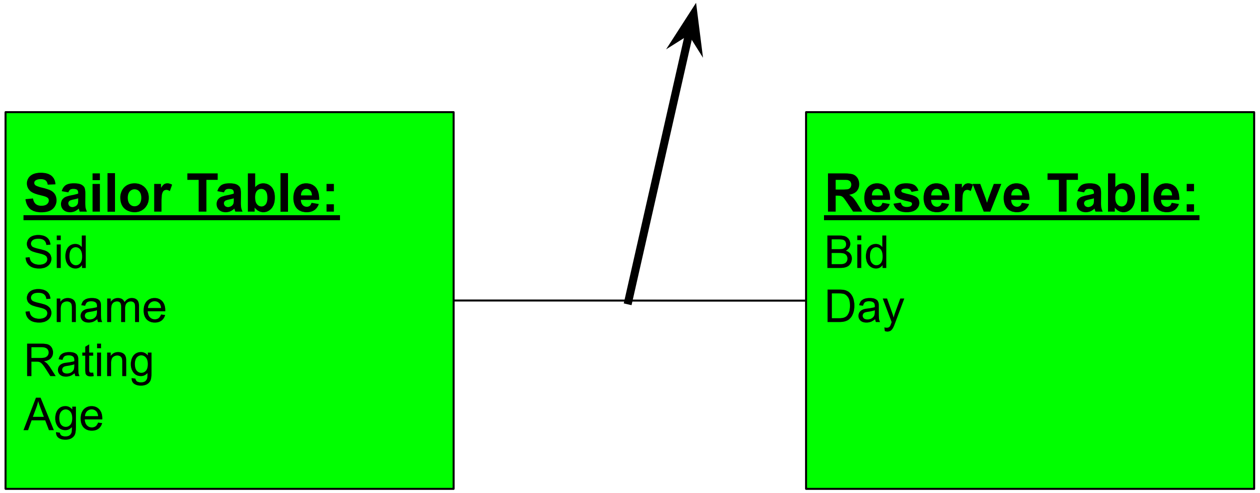
Bid
Day

3. Referential Integrity Constraints

Often we wish to ensure that a value appearing in a relation for a given set of attributes also appears for another set of attributes in another relation. This is called *referential integrity.*

Example:

Consider the Sailors relation and the constraint that no two sailors have the same SID. And, a sailor can't reserve a boat unless he/she is a valid sailor.



Sailor Table:
Sid
Sname
Rating
Age

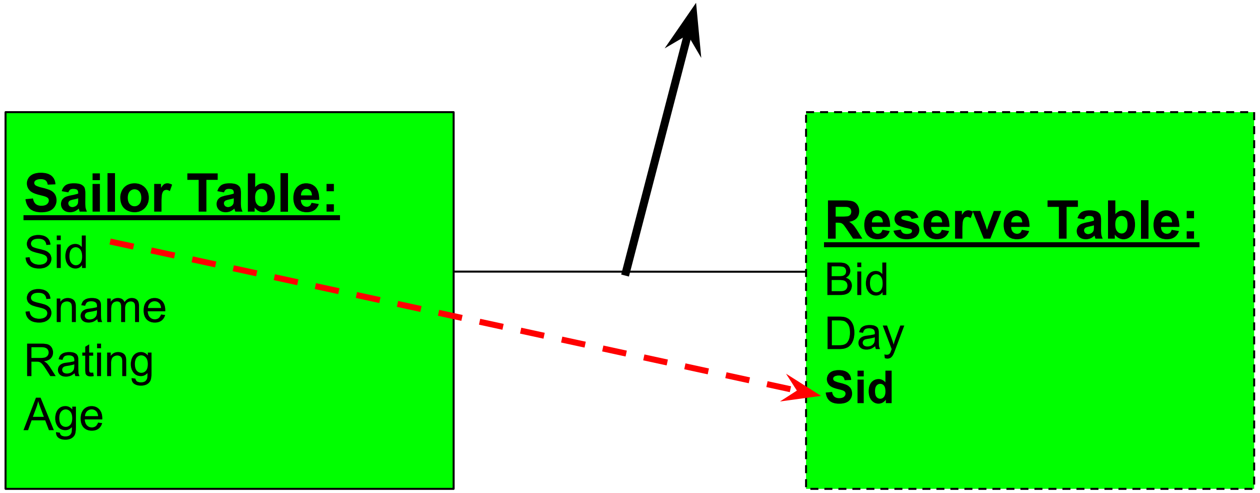
Reserve Table:
Bid
Day

3. Referential Integrity Constraints

Often we wish to ensure that a value appearing in a relation for a given set of attributes also appears for another set of attributes in another relation. This is called *referential integrity.*

Example:

Consider the Sailors relation and the constraint that no two sailors have the same SID. And, a sailor can't reserve a boat unless he/she is a valid sailor.

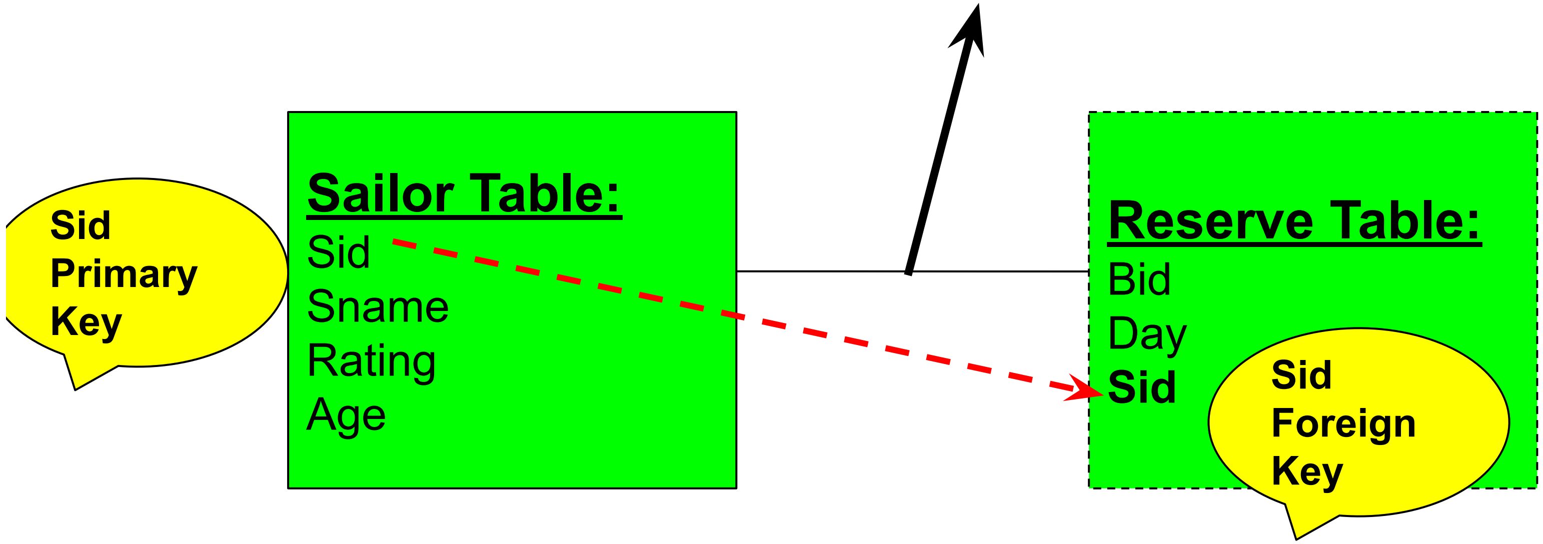


3. Referential Integrity Constraints

Often we wish to ensure that a value appearing in a relation for a given set of attributes also appears for another set of attributes in another relation. This is called *referential integrity.*

Example:

Consider the Sailors relation and the constraint that no two sailors have the same SID. And, a sailor can't reserve a boat unless he/she is a valid sailor.

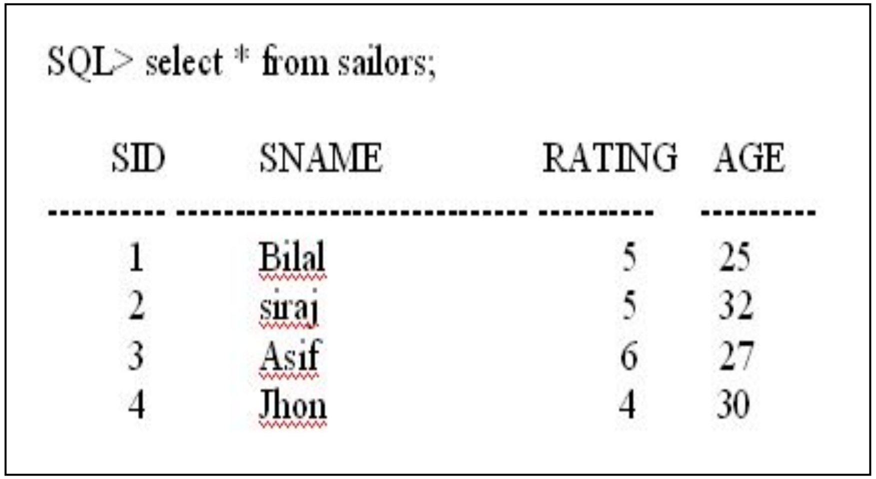


3. Referential Integrity Constraints

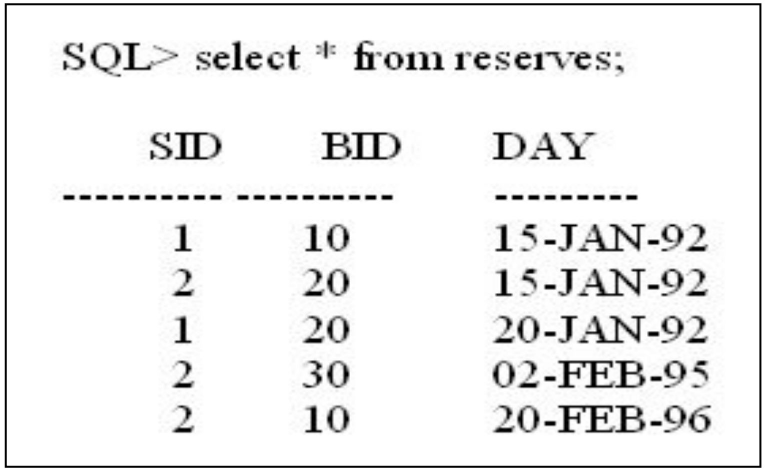
Often we wish to ensure that a value appearing in a relation for a given set of attributes also appears for another set of attributes in another relation. This is called *referential integrity.*

Example:

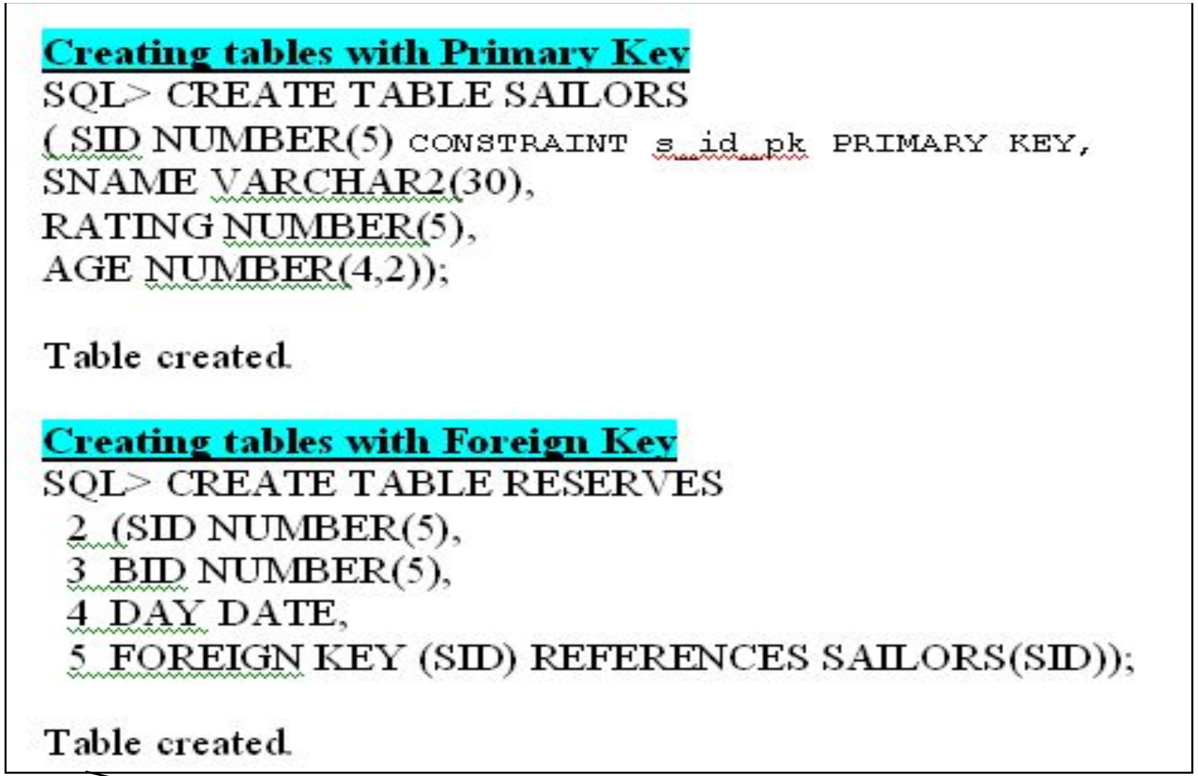
Consider the Sailors relation and the constraint that no two sailors have the same SID. And, a sailor can’t reserve a boat unless he/she is a valid sailor.



```
SQL> select * from sailors;

       SID  SNAME                             RATING    AGE
---------- ------------------------------ ----------  ----------
         1  Bilal                              5         25
         2  siraj                              5         32
         3  Asif                               6         27
         4  Jhon                               4         30
```



```
SQL> select * from reserves;

       SID        BID       DAY
---------- ----------       ---------
         1         10       15-JAN-92
         2         20       15-JAN-92
         1         20       20-JAN-92
         2         30       02-FEB-95
         2         10       20-FEB-96
```

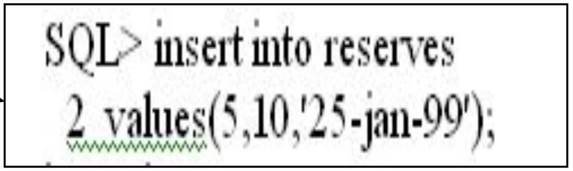


Creating tables with Primary Key

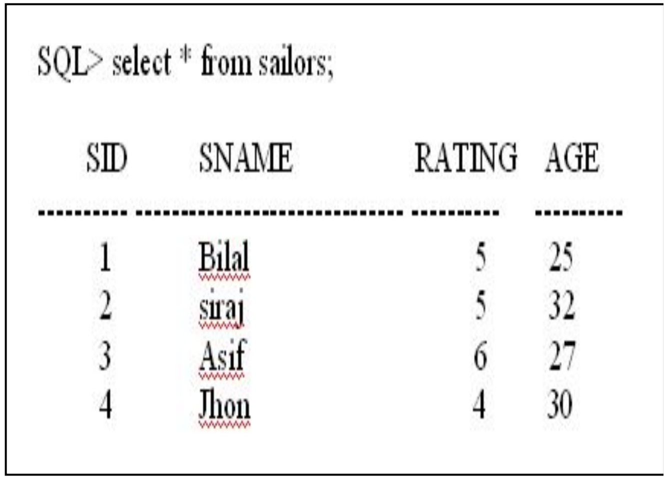
```
SQL> CREATE TABLE SAILORS
( SID NUMBER(5) CONSTRAINT s_id_pk PRIMARY KEY,
SNAME VARCHAR2(30),
RATING NUMBER(5),
AGE NUMBER(4,2));

Table created.
```

Creating tables with Foreign Key

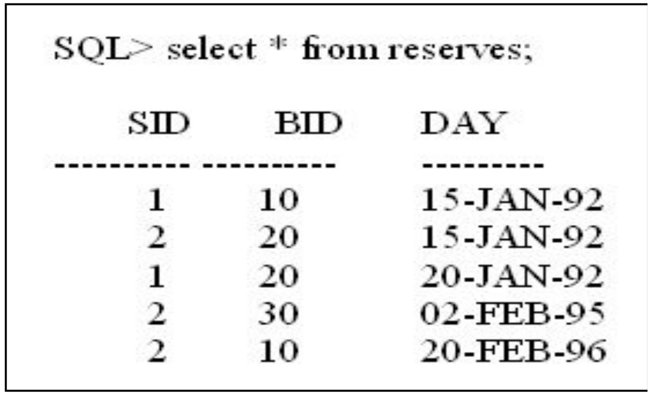
```
SQL> CREATE TABLE RESERVES
  2  (SID NUMBER(5),
  3  BID NUMBER(5),
  4  DAY DATE,
  5  FOREIGN KEY (SID) REFERENCES SAILORS(SID));

Table created.
```



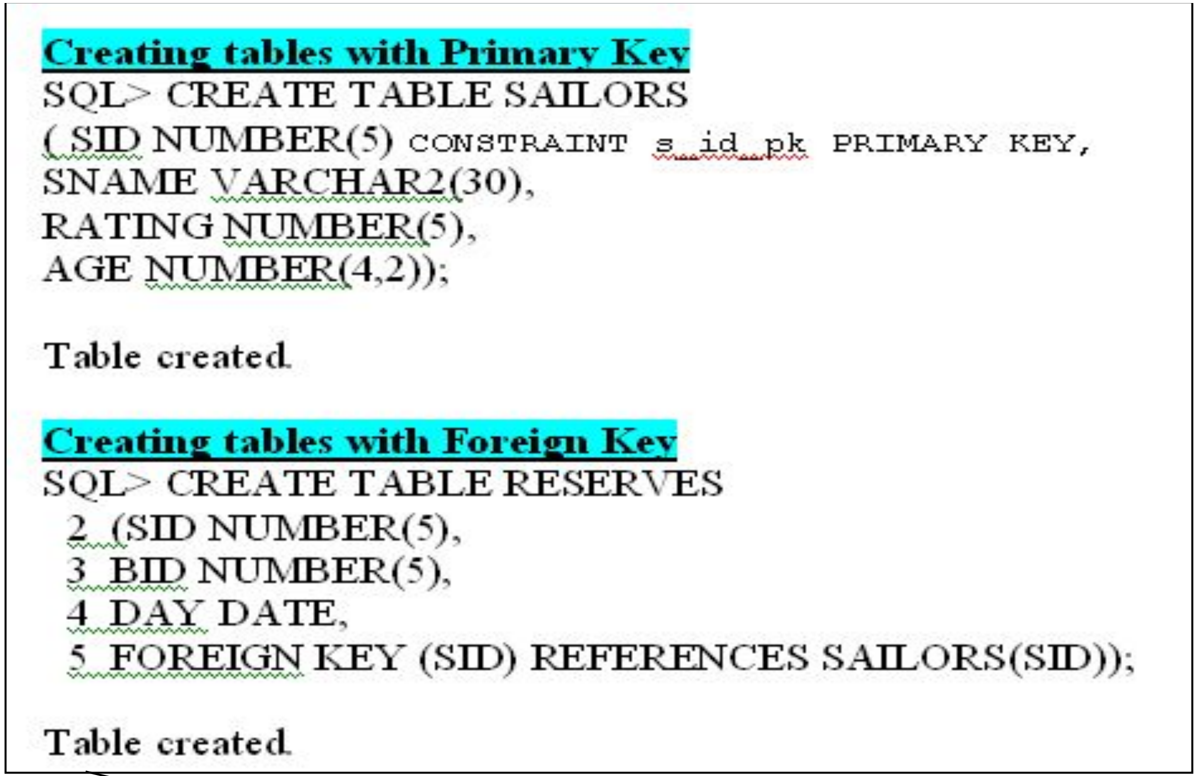
```
SQL> insert into reserves
  2  values(5,10,'25-jan-99');
```



```
SQL> select * from sailors;

       SID SNAME                              RATING        AGE
---------- ------------------------------ ---------- ----------
         1 Bilal                                   5         25
         2 siraj                                   5         32
         3 Asif                                    6         27
         4 Jhon                                    4         30
```



```
SQL> select * from reserves;

       SID        BID DAY
---------- ---------- ---------
         1         10 15-JAN-92
         2         20 15-JAN-92
         1         20 20-JAN-92
         2         30 02-FEB-95
         2         10 20-FEB-96
```

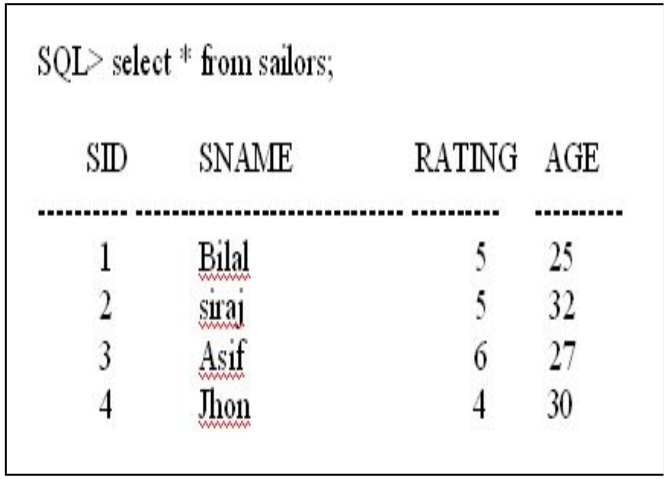


Creating tables with Primary Key

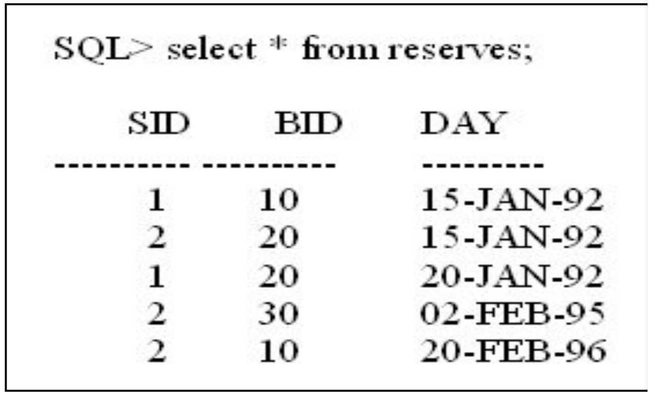
```
SQL> CREATE TABLE SAILORS
( SID NUMBER(5) CONSTRAINT s_id_pk PRIMARY KEY,
SNAME VARCHAR2(30),
RATING NUMBER(5),
AGE NUMBER(4,2));

Table created.
```

Creating tables with Foreign Key

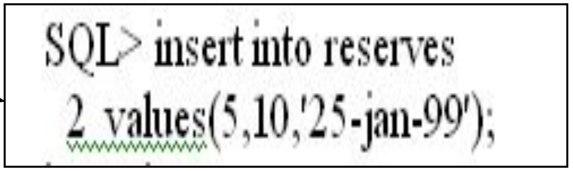
```
SQL> CREATE TABLE RESERVES
  2  (SID NUMBER(5),
  3  BID NUMBER(5),
  4  DAY DATE,
  5  FOREIGN KEY (SID) REFERENCES SAILORS(SID));

Table created.
```



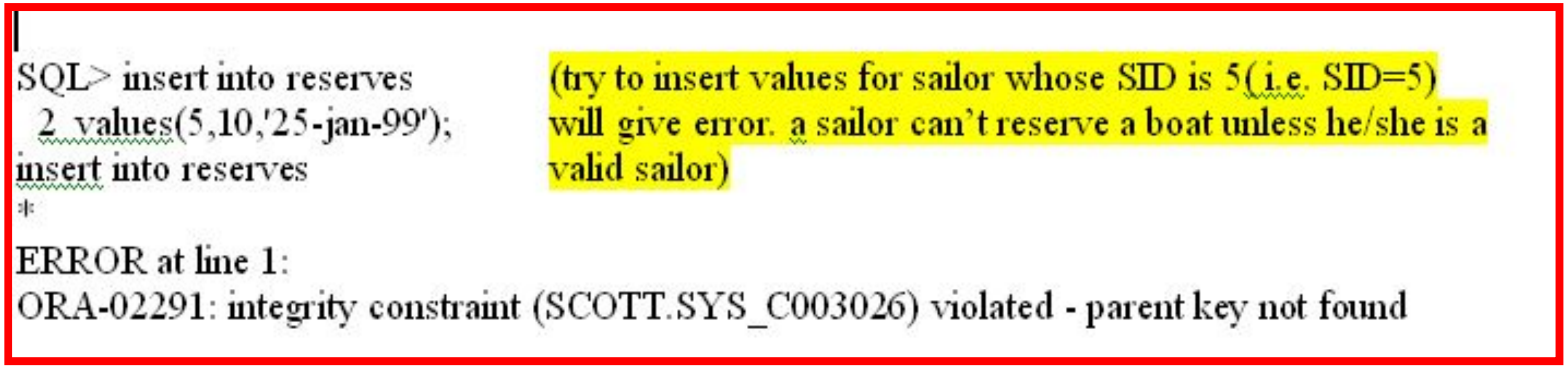
```
SQL> select * from sailors;

      SID     SNAME                          RATING     AGE
 --------- ----------------------------- ---------- ---------
         1    Bilal                             5      25
         2    siraj                             5      32
         3    Asif                              6      27
         4    Jhon                              4      30
```



```
SQL> select * from reserves;

       SID        BID        DAY
 ---------- ----------      ---------
         1         10        15-JAN-92
         2         20        15-JAN-92
         1         20        20-JAN-92
         2         30        02-FEB-95
         2         10        20-FEB-96
```



```
SQL> insert into reserves
  2  values(5,10,'25-jan-99');
```



```
SQL> insert into reserves
  2  values(5,10,'25-jan-99');
insert into reserves
*
ERROR at line 1:
ORA-02291: integrity constraint (SCOTT.SYS_C003026) violated - parent key not found
```

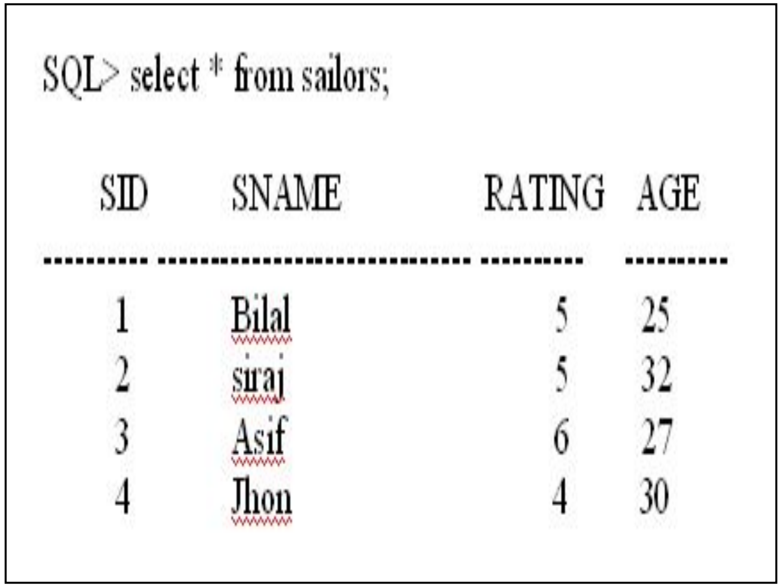
(try to insert values for sailor whose SID is 5(i.e. SID=5) will give error. a sailor can't reserve a boat unless he/she is a valid sailor)

Referential Actions

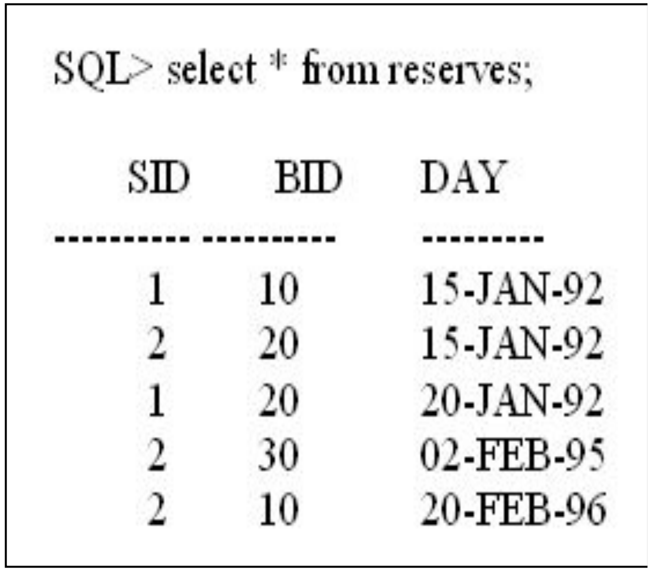
Because the DBMS enforces referential constraints, it must ensure data integrity if rows in a referenced table are to be deleted (or updated). If dependent rows in referencing tables still exist, those references have to be considered.

SQL: 2003 specifies 5 different **referential actions** that shall take place in such occurrences:

1. CASCADE
2. RESTRICT
3. NO ACTION
4. SET NULL
5. SET DEFAULT



```
SQL> select * from sailors;

       SID SNAME                              RATING        AGE
---------- ------------------------------ ---------- ----------
         1 Bilal                                   5         25
         2 siraj                                   5         32
         3 Asif                                    6         27
         4 Jhon                                    4         30
```



```
SQL> select * from reserves;

       SID        BID DAY
---------- ---------- ---------
         1         10 15-JAN-92
         2         20 15-JAN-92
         1         20 20-JAN-92
         2         30 02-FEB-95
         2         10 20-FEB-96
```

Discussion Questions

Figure 3.6
One possible database state for the COMPANY relational database schema.

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

DEPENDENT

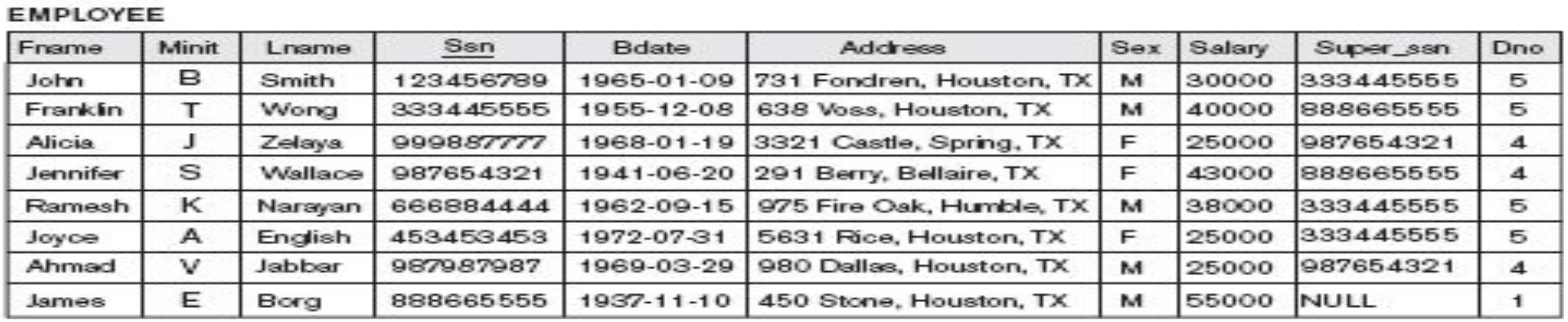
Essn	Dependent_name	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

Dealing with Constraint Violations

I.The Insert Operation

Operation 1:

Insert <‘Cecilia’, ‘F’, ‘Kolonsky’, NULL, ‘1960-04-05’, ‘6357 Windy Lane, Katy, TX’, F, 28000, NULL, 4> into EMPLOYEE.



EMPLOYEE

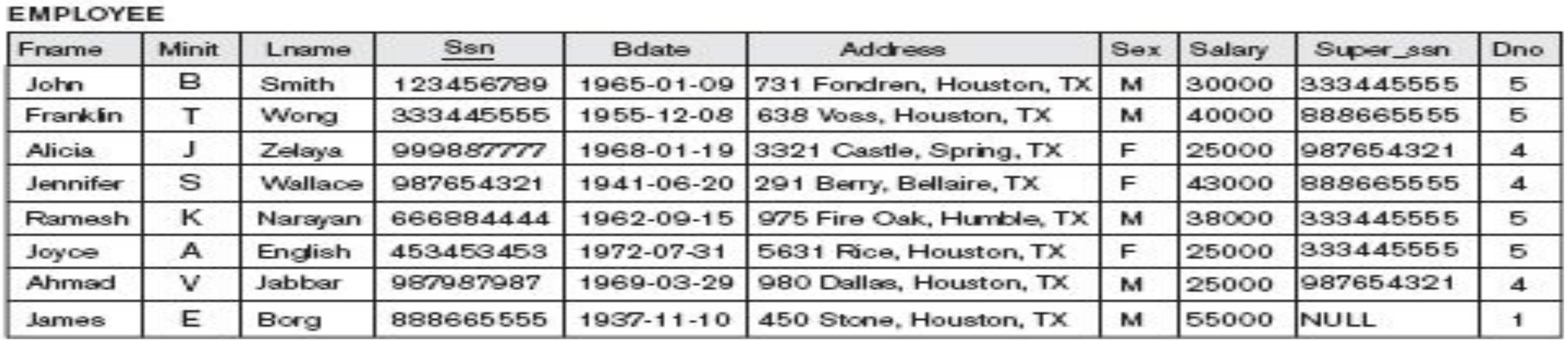
Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

Dealing with Constraint Violations

I.The Insert Operation

Operation 1:

Insert <‘Cecilia’, ‘F’, ‘Kolonsky’, **NULL**, ‘1960-04-05’, ‘6357 Windy Lane, Katy, TX’, F, 28000, NULL, 4> into EMPLOYEE.



EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

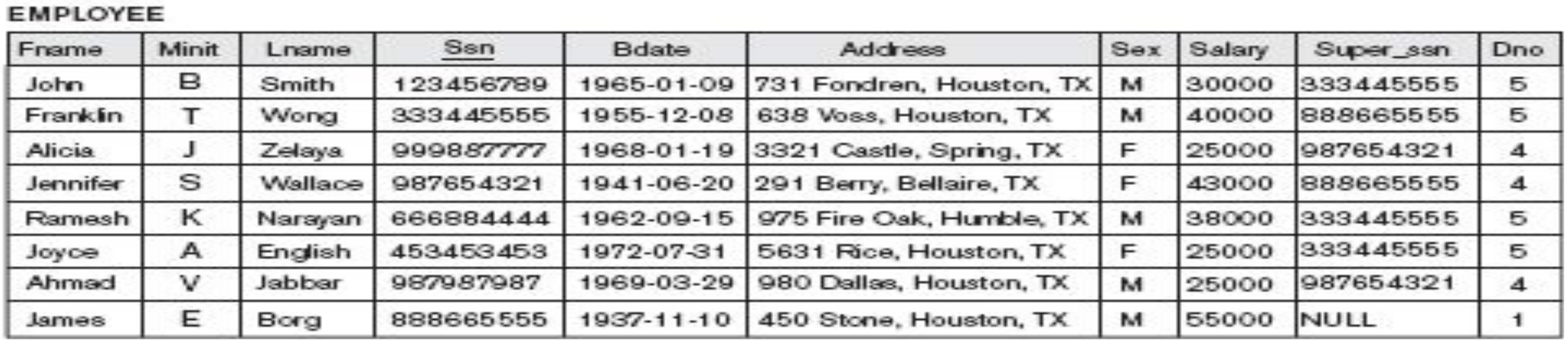
Result: This insertion violates the entity integrity constraint (NULL for the primary key Ssn), so it is rejected.

Dealing with Constraint Violations

I.The Insert Operation

Operation 2:

Insert <‘Alicia’, ‘J’, ‘Zelaya’, ‘999887777’, ‘1960-04-05’, ‘6357 Windy Lane, Katy, TX’, F, 28000, ‘987654321’, 4> into EMPLOYEE.



EMPLOYEE

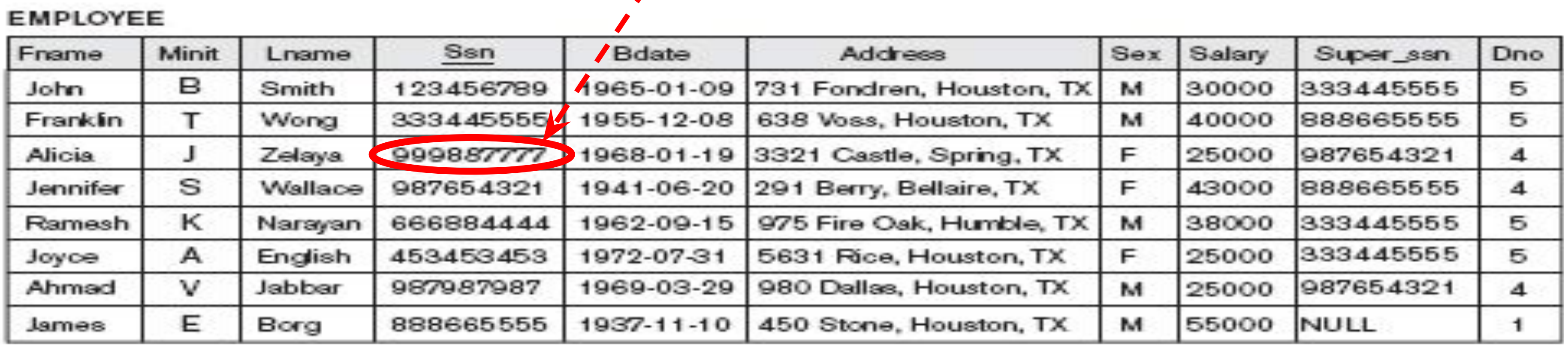
Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

Dealing with Constraint Violations

I.The Insert Operation

Operation 2:

Insert <'Alicia', 'J', 'Zelaya', '999887777', '1960-04-05', '6357 Windy Lane, Katy, TX', F, 28000, '987654321', 4> into EMPLOYEE.



EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

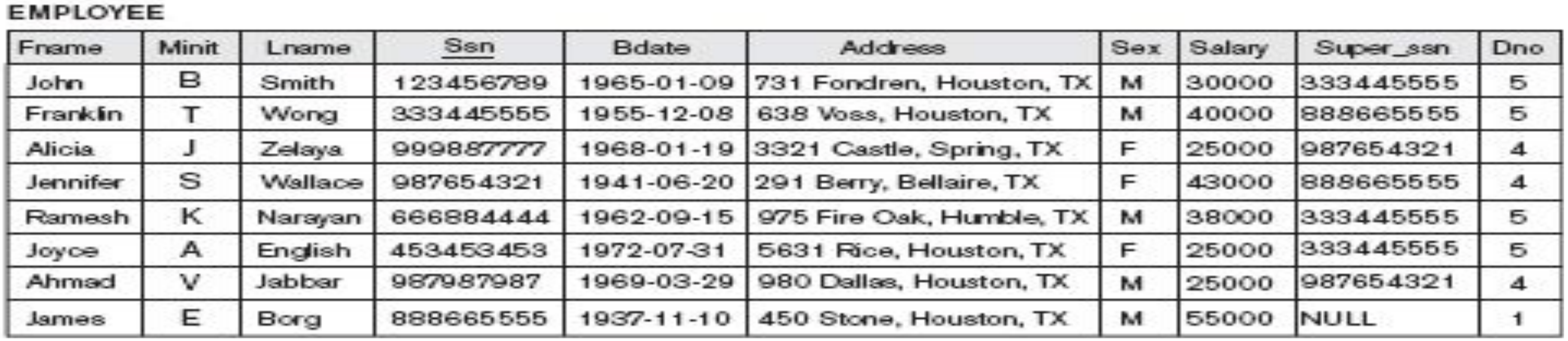
Result: This insertion violates the key constraint because another tuple with the same Ssn value already exists in the EMPLOYEE relation, and so it is rejected.

Dealing with Constraint Violations

I.The Insert Operation

Operation 3:

Insert <'Cecilia', 'F', 'Kolonsky', '677678989', '1960-04-05', '6357 Windswept, Katy, TX', F, 28000, '987654321', 7> into EMPLOYEE.



EMPLOYEE

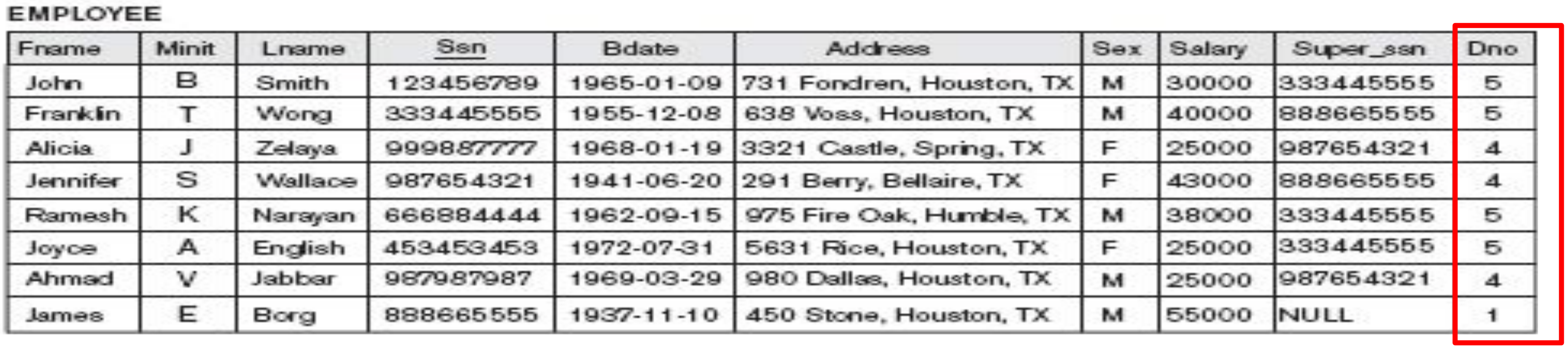
Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

Dealing with Constraint Violations

I.The Insert Operation

Operation 3:

Insert <‘Cecilia’, ‘F’, ‘Kolonsky’, ‘677678989’, ‘1960-04-05’, ‘6357 Windswept, Katy, TX’, F, 28000, ‘987654321’, 7> into EMPLOYEE.



EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

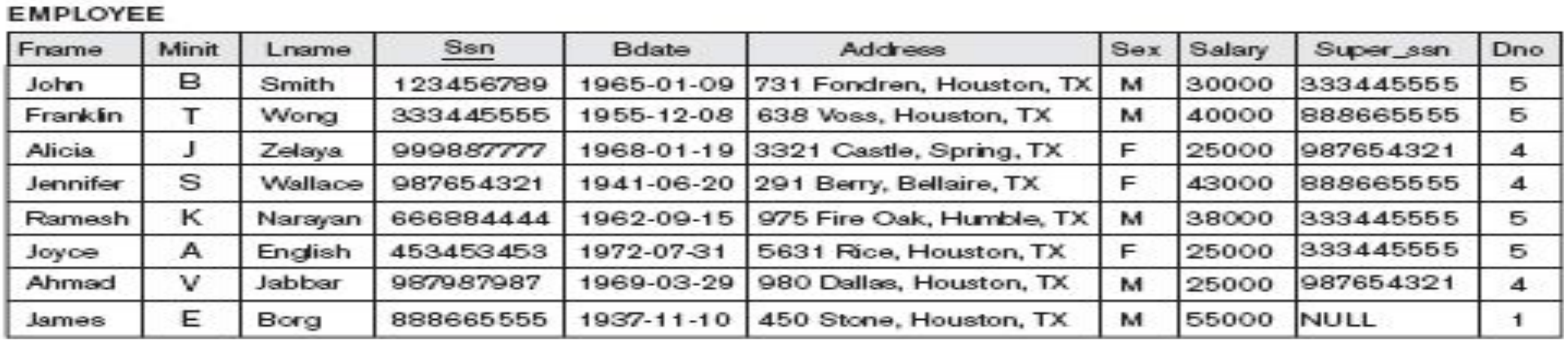
Result: This insertion violates the referential integrity constraint specified on Dno in EMPLOYEE because no corresponding referenced tuple exists in DEPARTMENT with Dnumber = 7.

Dealing with Constraint Violations

I.The Insert Operation

Operation 4:

Insert <‘Cecilia’, ‘F’, ‘Kolonsky’, ‘677678989’, ‘1960-04-05’, ‘6357 Windy Lane, Katy, TX’, F, 28000, NULL, 4> into EMPLOYEE.



EMPLOYEE

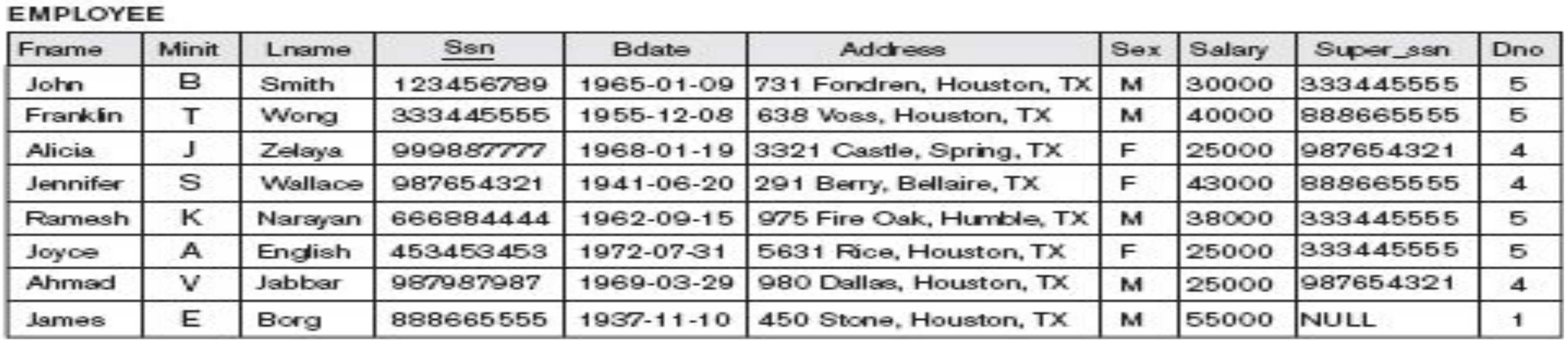
Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

Dealing with Constraint Violations

I.The Insert Operation

Operation 4:

Insert <‘Cecilia’, ‘F’, ‘Kolonsky’, ‘677678989’, ‘1960-04-05’, ‘6357 Windy Lane, Katy, TX’, F, 28000, NULL, 4> into EMPLOYEE.



EMPLOYEE

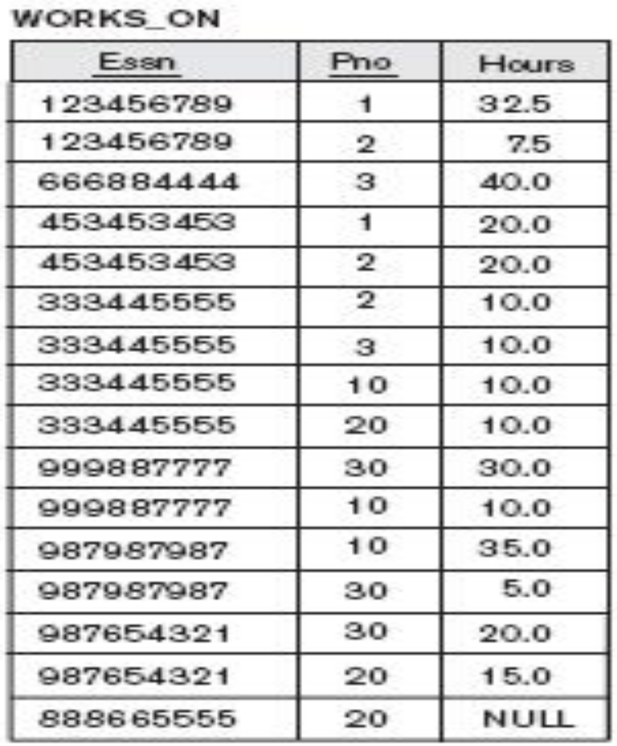
Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

Result: This insertion satisfies all constraints, so it is acceptable.

Dealing with Constraint Violations

II. The Delete Operation

Operation 5: Delete the WORKS_ON tuple with Essn = ‘999887777’ and Pno = 10.



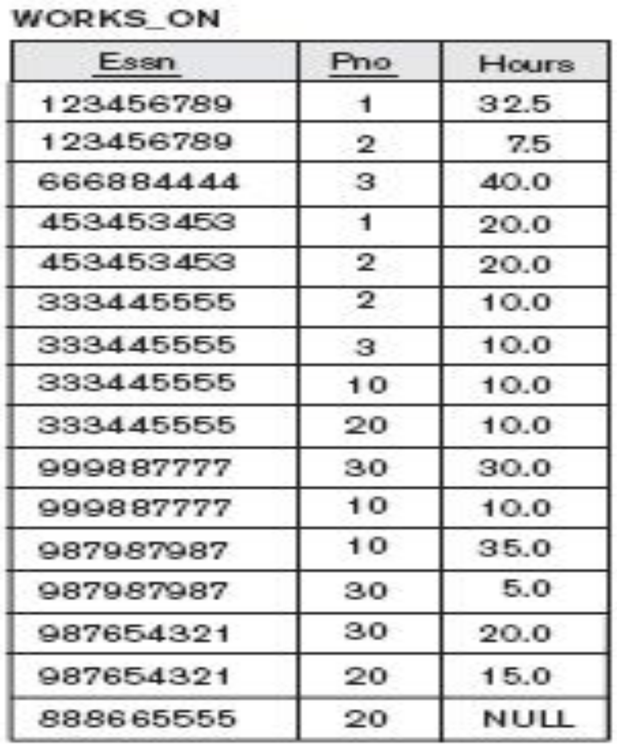
WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

Dealing with Constraint Violations

II. The Delete Operation

Operation 5: Delete the WORKS_ON tuple with Essn = ‘999887777’ and Pno = 10.



WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

Result: This deletion is acceptable and deletes exactly one tuple.

Dealing with Constraint Violations

II. The Delete Operation

Operation 6:

Delete the EMPLOYEE tuple with Ssn = ‘999887777’.

Dealing with Constraint Violations

II. The Delete Operation

Operation 6:

Delete the EMPLOYEE tuple with Ssn = ‘999887777’.

Result: This deletion is not acceptable, because there are tuples in WORKS_ON that refer to this tuple. Hence, if the tuple in EMPLOYEE is deleted, referential integrity violations will result.

Dealing with Constraint Violations

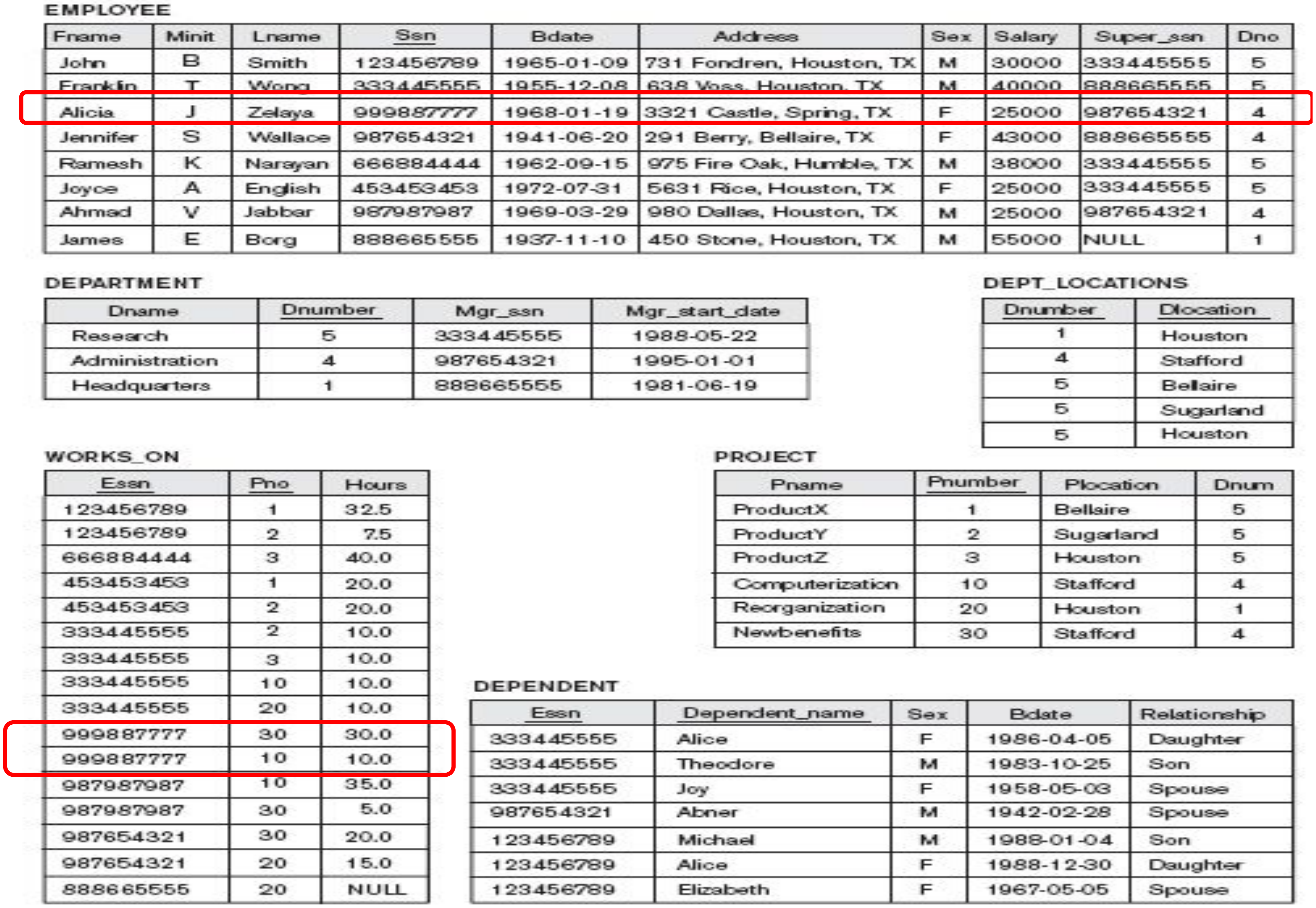
II. The Delete Operation

Operation 6:

Delete the EMPLOYEE tuple with Ssn = ‘999887777’.

Result: This deletion is not acceptable, because there are tuples in WORKS_ON that refer to this tuple. Hence, if the tuple in EMPLOYEE is deleted, referential integrity violations will result.

Operation 6: Delete the EMPLOYEE tuple with Ssn = ‘999887777’.



EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

Dealing with Constraint Violations

II. The Delete Operation

Operation 7:

Delete the EMPLOYEE tuple with Ssn = ‘333445555’.

Dealing with Constraint Violations

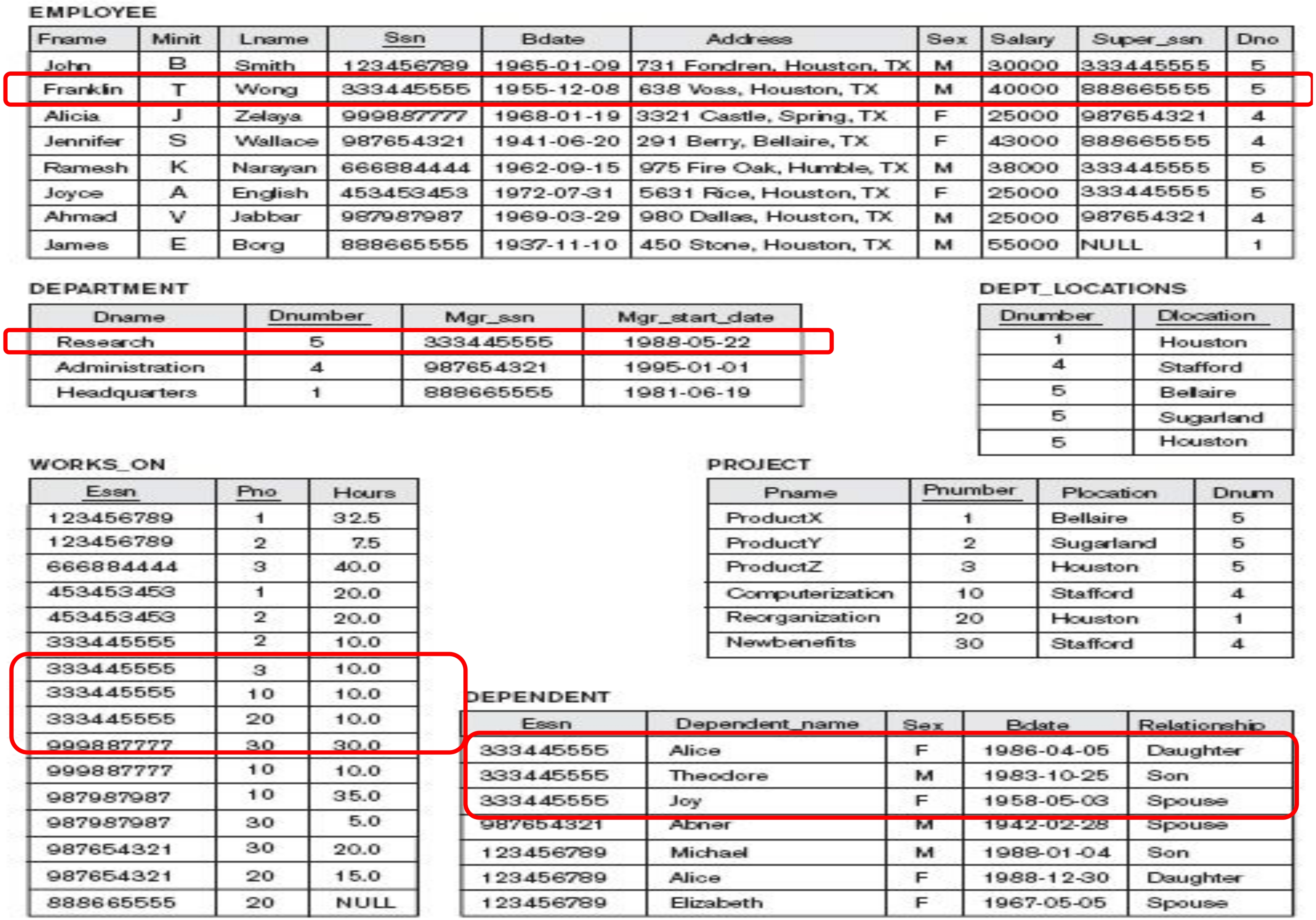
II. The Delete Operation

Operation 7:

Delete the EMPLOYEE tuple with Ssn = ‘333445555’.

Result: This deletion will result in even worse referential integrity violations, because the tuple involved is referenced by tuples from the EMPLOYEE, DEPARTMENT,WORKS_ON, and DEPENDENT relations.

Operation7: Delete the EMPLOYEE tuple with Ssn = ‘333445555’.



EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

Dealing with Constraint Violations

III. The Update Operation

Operation 8:

Update the salary of the EMPLOYEE tuple with Ssn = ‘999887777’ to 28000.

Dealing with Constraint Violations

III. The Update Operation

Operation 8:

Update the salary of the EMPLOYEE tuple with Ssn = ‘999887777’ to 28000.

Result: Acceptable.

Dealing with Constraint Violations

III. The Update Operation

Operation 8:

Update the salary of the EMPLOYEE tuple with Ssn = ‘999887777’ to 28000.

Result: Acceptable.

Operation 9:

Update the Dno of the EMPLOYEE tuple with Ssn = ‘999887777’ to 1.

Dealing with Constraint Violations

III. The Update Operation

Operation8:

Update the salary of the EMPLOYEE tuple with Ssn = ‘999887777’ to 28000.

Result: Acceptable.

Operation 9:

Update the Dno of the EMPLOYEE tuple with Ssn = ‘999887777’ to 1.

Result: Acceptable.

Dealing with Constraint Violations

III. The Update Operation

Operation 8:

Update the salary of the EMPLOYEE tuple with Ssn = ‘999887777’ to 28000.

Result: Acceptable.

Operation 9:

Update the Dno of the EMPLOYEE tuple with Ssn = ‘999887777’ to 1.

Result: Acceptable.

Operation 10:

Update the Dno of the EMPLOYEE tuple with Ssn = ‘999887777’ to 7.

Dealing with Constraint Violations

III. The Update Operation

Operation 8:

Update the salary of the EMPLOYEE tuple with Ssn = ‘999887777’ to 28000.

Result: Acceptable.

Operation 9:

Update the Dno of the EMPLOYEE tuple with Ssn = ‘999887777’ to 1.

Result: Acceptable.

Operation 10:

Update the Dno of the EMPLOYEE tuple with Ssn = ‘999887777’ to 7.

Result: Unacceptable, because it violates referential integrity.

Dealing with Constraint Violations

III. The Update Operation

Operation 8:

Update the salary of the EMPLOYEE tuple with Ssn = ‘999887777’ to 28000.

Result: Acceptable.

Operation 9:

Update the Dno of the EMPLOYEE tuple with Ssn = ‘999887777’ to 1.

Result: Acceptable.

Operation 10:

Update the Dno of the EMPLOYEE tuple with Ssn = ‘999887777’ to 7.

Result: Unacceptable, because it violates referential integrity.

Operation 11:

Update the Ssn of the EMPLOYEE tuple with Ssn = ‘999887777’ to ‘987654321’.

Dealing with Constraint Violations

III. The Update Operation

Operation 8:

Update the salary of the EMPLOYEE tuple with Ssn = ‘999887777’ to 28000.

Result: Acceptable.

Operation 9:

Update the Dno of the EMPLOYEE tuple with Ssn = ‘999887777’ to 1.

Result: Acceptable.

Operation 10:

Update the Dno of the EMPLOYEE tuple with Ssn = ‘999887777’ to 7.

Result: Unacceptable, because it violates referential integrity.

Operation 11:

Update the Ssn of the EMPLOYEE tuple with Ssn = ‘999887777’ to ‘987654321’.

Result: Unacceptable, because it violates primary key constraint by repeating a value that already exists as a primary key in another tuple; it violates referential integrity constraints because there are other relations that refer to the existing value of Ssn.