CSE 302
Database Management Systems Sessional

Constraints

Why do we need constraints?

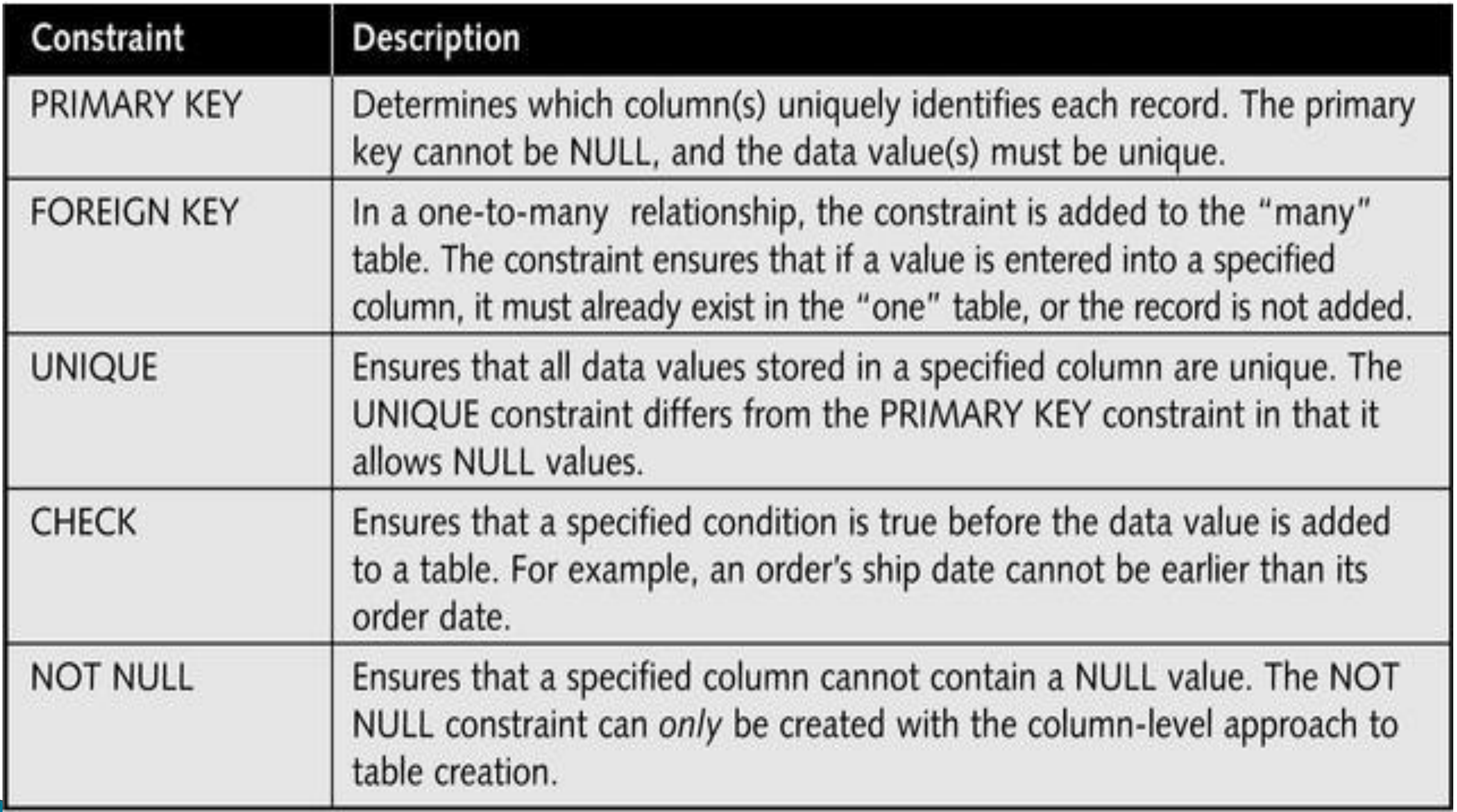
- To keep the database reliable.
- To prevent a user from entering non-sensible data.
- The business or other organization has certain rules that cannot be violated.
- Constraints are used for implementing the rules.

We can use the constraints for the following reasons:

1. Enforce rules at the table level whenever a row is **inserted, updated or deleted** from the table. The constraints must be satisfied for the operation to be succeed.

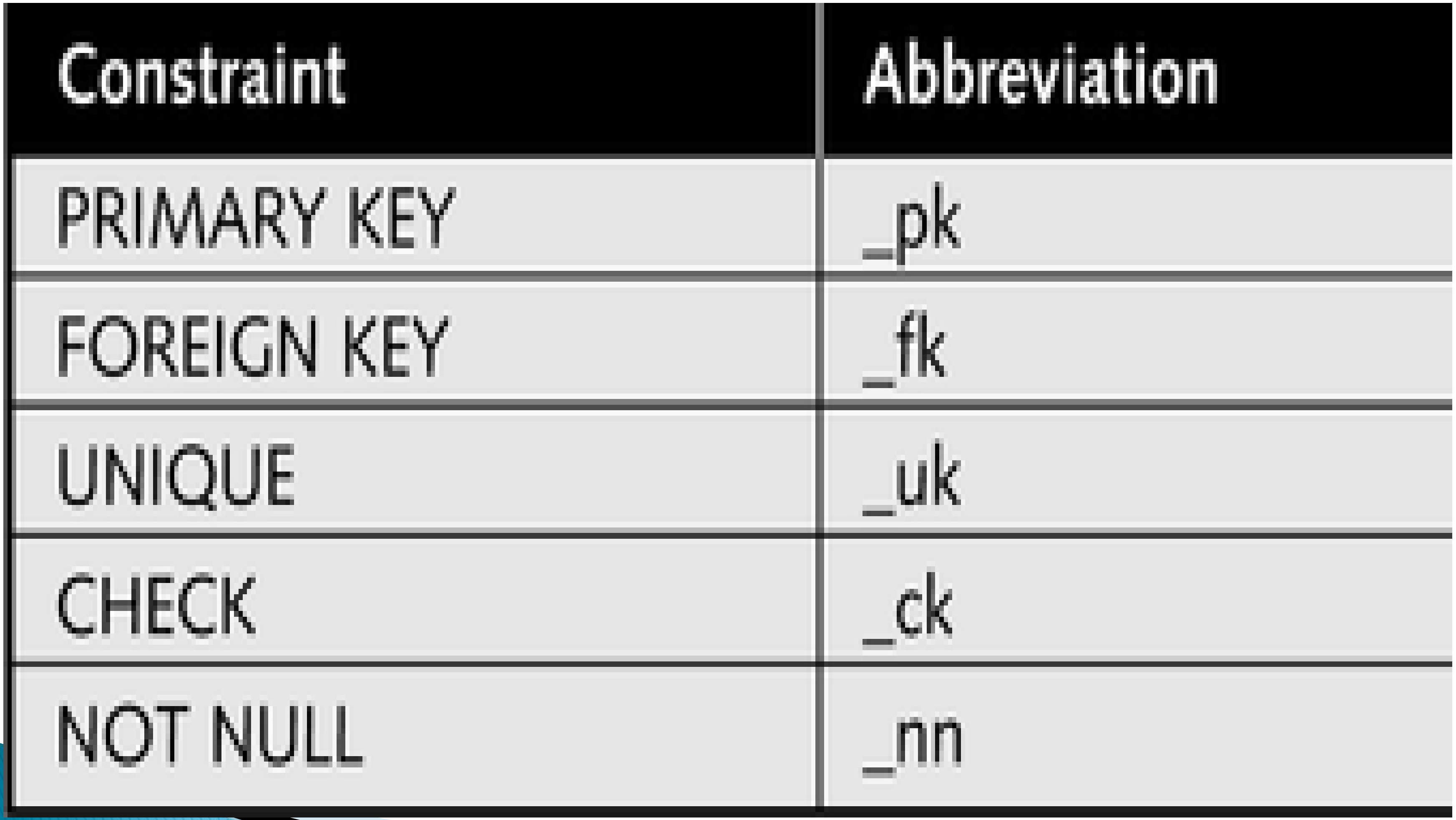
2. Prevent the **deletion** of a table if there are **dependencies** from other tables.

Types of constraints



Constraint	Description
PRIMARY KEY	Determines which column(s) uniquely identifies each record. The primary key cannot be NULL, and the data value(s) must be unique.
FOREIGN KEY	In a one-to-many relationship, the constraint is added to the “many” table. The constraint ensures that if a value is entered into a specified column, it must already exist in the “one” table, or the record is not added.
UNIQUE	Ensures that all data values stored in a specified column are unique. The UNIQUE constraint differs from the PRIMARY KEY constraint in that it allows NULL values.
CHECK	Ensures that a specified condition is true before the data value is added to a table. For example, an order’s ship date cannot be earlier than its order date.
NOT NULL	Ensures that a specified column cannot contain a NULL value. The NOT NULL constraint can *only* be created with the column-level approach to table creation.

ABBREVIATION



Constraint	Abbreviation
PRIMARY KEY	_pk
FOREIGN KEY	_fk
UNIQUE	_uk
CHECK	_ck
NOT NULL	_nn

Ways of applying Constraints

Constraints can be applied in two ways:

As part of a CREATE TABLE command

or

As part of an ALTER TABLE command

Syntax for entering a constraint name

Use the following syntax for entering a constraint name:

Tablename_Columnname_ConstraintType

```
ALTER TABLE CUSTOMER
ADD CONSTRAINT CUSTOMER_CUST_ID_pk
PRIMARY KEY (CUST_ID)
```

$\underbrace{\text{CUSTOMER}}_{\text{Table Name}}\_\underbrace{\text{CUST\_ID}}_{\text{Column Name}}\_\underbrace{\text{pk}}_{\text{Constraint Type}}$

PRIMARY KEY

```
ALTER TABLE CUSTOMER ADD CONSTRAINT
 Customer_CUST_ID_pk PRIMARY KEY(CUST_ID);


Create table Customer (
        Cust_id VARCHAR2(12) ,
         Cust_nam VARCHAR2(12),
         Cust_dob DATE,
         Cust_street VARCHAR2(12),
        Cust_city VARCHAR2(12),
CONSTRAINT Customer_CUST_ID_pk      PRIMARY
KEY(CUST_ID)
                              );
```

```
Create table Customer
(
    Cust_id VARCHAR2(12) PRIMARY KEY,
    Cust_nam VARCHAR2(12),
    Cust_dob DATE,
    Cust_street VARCHAR2(12),
    Cust_city VARCHAR2(12),
);
```

```
Create table Customer
(
    Cust_id VARCHAR2(12) ,
    Cust_nam VARCHAR2(12),
    Cust_dob DATE,
    Cust_street VARCHAR2(12),
    Cust_city VARCHAR2(12),
    CONSTRAINT Customer_CUST_ID_pk PRIMARY KEY(CUST_ID)
);
```

```
ALTER TABLE CUSTOMER
ADD CONSTRAINT Customer_CUST_ID_pk PRIMARY KEY(CUST_ID);
```

PRIMARY KEY – COMPOSITE

- Simply list the column names within parentheses after the constraint type.

```
ALTER TABLE orderitems
ADD CONSTRAINT orderitems_pk PRIMARY KEY (order#, item#);
```

Table Name

Constraint Type

Multiple Column Names

- After this constraint is added to the ORDERITEMS table, a user can enter only a unique combination of Order# and Item# for each new row.

NOT NULL

```
Create table Customer
(
  Cust_id  VARCHAR2(12)  NOT NULL,
  Cust_name      VARCHAR2(12),
  Cust_dob       DATE,
  Cust_street    VARCHAR2(12),
  Cust_city      VARCHAR2(12)
);

ALTER TABLE CUSTOMER
MODIFY (CUST_ID NOT NULL);
```

UNIQUE Constraints

```
Create table Account
(
Account_id VARCHAR2(12) NOT NULL ,
Balance NUMBER(20,5),
Type VARCHAR2(8),
CONSTRAINT Account_ACCID_uk UNIQUE(Account_id)
);
```

```
ALTER TABLE ACCOUNT
ADD CONSTRAINT ACCOUNT_ACCOUNT_ID_uk
   UNIQUE(ACCOUNT_ID);
```

UNIQUE

- A **UNIQUE** constraint allows **NULL** values unless define **NOT NULL** in the same column
- A **PRIMARY KEY** constraint does not allow **NULL** values

FOREIGN KEY constraint

```
ALTER TABLE Depositor
ADD CONSTRAINT Depositor_Cust_ID_fk
FOREIGN KEY (Cust_ID) REFERENCES
Customer (Cust_ID);
```

FOREIGN KEY Composite

```
CREATE TABLE Depositor (
Cust_id VARCHAR2(12) NOT NULL,
 Account_id VARCHAR2(12) NOT NULL,
  COSNTRAINT DEPOSITOR_CUST_ID_FK
  FOREIGN KEY(CUST_ID) REFERENCES
  CUSTOMER(CUST_ID),
  COSNTRAINT DEPOSITOR_ACCOUNT_ID_FK
  FOREIGN KEY(ACCOUNT_ID) REFERENCES
  ACCOUNT(ACCOUNT_ID)
                    );
```

Foreign Key

- A record cannot be deleted in the parent table (CUSTOMER) if matching entries exist in the child table.

  ALTER TABLE DEPOSITOR ADD CONSTRAINT DEPOSITOR _CUST_ID_fk FOREIGN KEY (CUST_ID) REFERENCES CUSTOMER (CUST_ID) ON DELETE CASCADE;

- If a record is deleted from the parent table, then any corresponding records in the child table are also automatically deleted.

CHECK Constraints

```
Create table Account
(
Account_id VARCHAR2(12) NOT NULL UNIQUE,
Balance NUMBER(20,5) CHECK( Balance>0),
Type VARCHAR2(8)
);

Create table Account
(
Account_id VARCHAR2(12) NOT NULL UNIQUE,
Balance NUMBER(20,5),
Type VARCHAR2(8)
CONSTRAINT Account_Balance_ck  CHECK(Balance>0)
);
```

ADD Constraints

- You can add, drop, enable or disable a constraint, but you cannot modify its structure.
- You can add a NOT NULL constraint to an existing column by using the MODIFY Clause of the ALTER TABLE statement.

DROP Constraints

- To drop a constraint, you can identify the constraint name from the USER_CONSTRAINTS and then use ALTER TABLE command with the DROP clause.

- To remove the primary key constraint from the Customer Table and drop the associated FOREIGN KEY constraint–

```
ALTER TABLE CUSTOMER
DROP PRIMARY KEY CASCADE;
```

Viewing constraints

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

```
SELECT CONSTRAINT_NAME, CONSTRAINT_TYPE,
   SEARCH_CONDITION
FROM USER_CONSTRAINTS
WHERE TABLE_NAME=‘CUSTOMER’;
```

- *Viewing The Columns Associated With Constraints*

```
SELECT CONSTRAINT_NAME,COLUMN_NAME
FROM USER_CONS_COLUMNS
WHERE TABLE_NAME=’CUSTOMER’;
```

Practice Problems
» for
Constraints

- Add a FOREIGN KEY CONSTRAINT on the EMPLOYEE table that ensures that each Employee's Manager also exists in Employee Table.

- CREATE TABLE BORROWER in such a way that Cust_ID must be in Customer table and Loan_ID must be in LOAN table.

THANK YOU