

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

شروع اللہ کے پاک نام سے جو بڑا مہربان نہایت رحم والا ہے



Dr. Abid Sohail Bhutta

abidbhutta@cuilahore.edu.pk

**Department of Computer
Science,**

**COMSATS University , Lahore
Campus**

Database Systems

Lecture 10

SQL Constraints

Foreign key, Unique, Not Null and Check constraints in SQL



Recall Lecture 9

- Relational DBMS
- Relation/Table in Relational Model
- Keys
 - Primary Key
 - Unique Key
 - Surrogate Key

Relational Integrity

Constraints placed on the set of values allowed for the attributes of a relation.

- Entity integrity

- No attribute of a primary key can be null (every tuple must be uniquely identified)

- Referential integrity

- If a foreign key exists in a relation, either the foreign key value must match a candidate key value of some tuple in its home relation, or the foreign key value must be wholly null (i.e., no key exists in the home relation)

- Enterprise constraints (organizational)

SQL Constraints

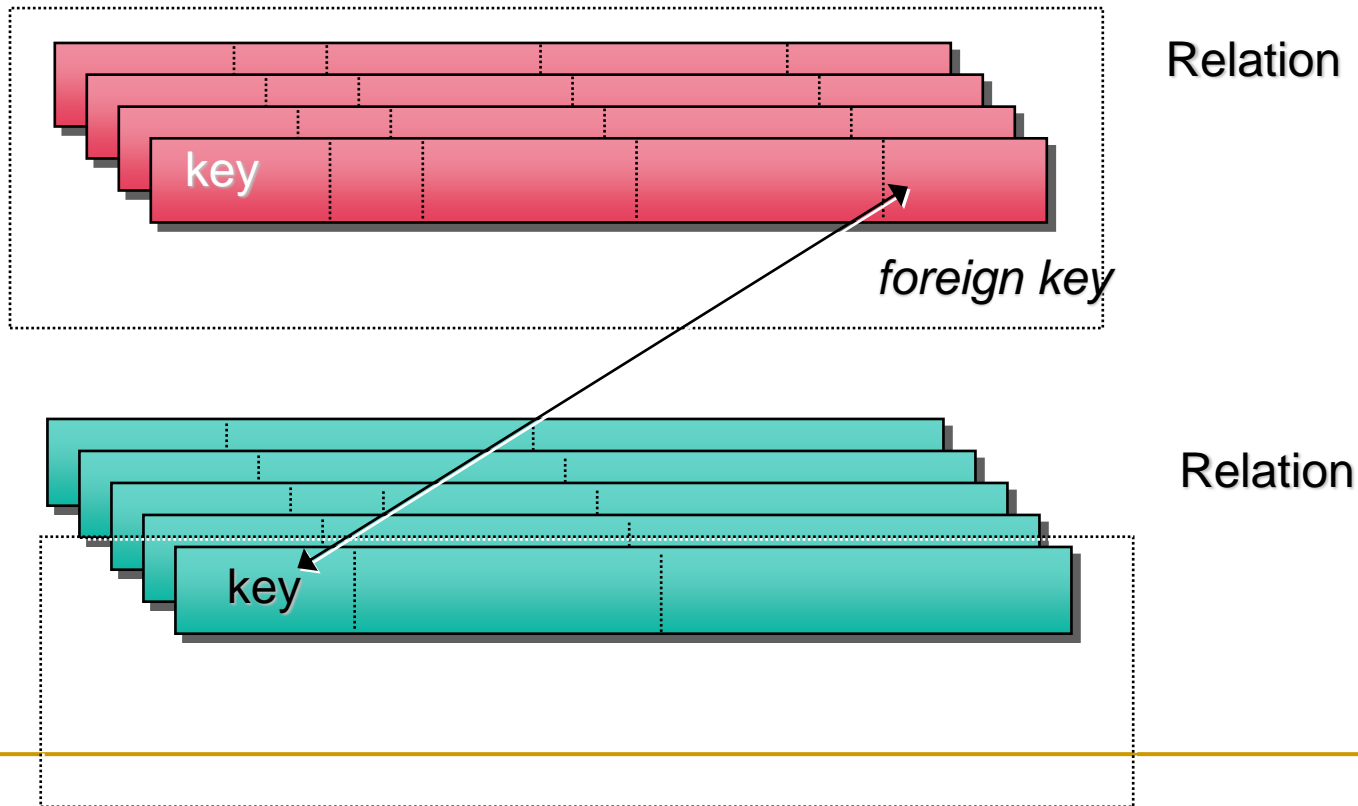
- SQL constraints are used to specify rules for the data in a table.
- Constraints are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the table. If there is any violation between the constraint and the data action, the action is aborted.
- Constraints can be column level or table level. Column level constraints apply to a column, and table level constraints apply to the whole table.

SQL Constraints

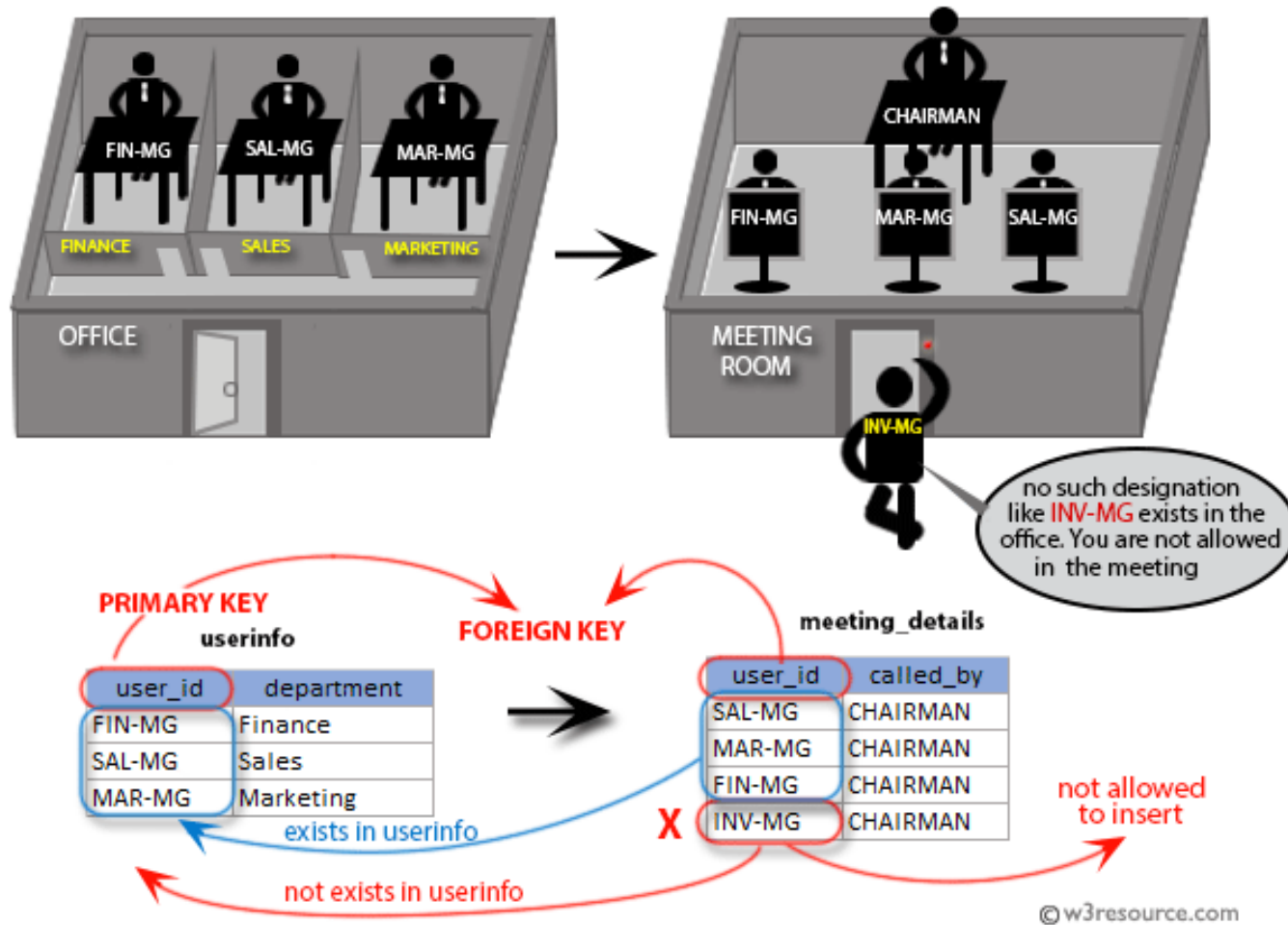
- NOT NULL
 - Ensures that a column cannot have a NULL value
- UNIQUE
 - Ensures that all values in a column are different
- PRIMARY KEY
 - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table
- FOREIGN KEY
 - Uniquely identifies a row/record in another table
- CHECK
 - Ensures that all values in a column satisfies a specific condition
- DEFAULT
 - Sets a default value for a column when no value is specified
- INDEX
 - Used to create and retrieve data from the database very quickly

Foreign key

An attribute or set of attributes within one relation that matches the candidate key of some (possibly the same) relation



Foreign Key (FK) Example



Foreign Key (FK) Example

■ Table Person

PersonID	LastName	FirstName	Age
1	Hansen	Ola	30
2	Svendson	Tove	23
3	Pettersen	Kari	20

■ Table Orders

OrderID	OrderNumber	PersonID
1	77895	3
2	44678	3
3	22456	2
4	24562	1

```
CREATE TABLE Orders (  
    OrderID int NOT NULL PRIMARY KEY,  
    OrderNumber int NOT NULL,  
    PersonID int FOREIGN KEY REFERENCES Persons(PersonID  
);
```

Foreign Key (FK) Example SQL Code

Use Exapmle

Drop table Person

Drop table Orders

Select * from Person

Select * from Orders

create table Person (

 PersonId int ,

 FirstName nvarchar(40) not null,

 LastName nvarchar(40) not null,

 age nvarchar(40),

 constraint PK_Person primary key (PersonId));

CREATE TABLE Orders (

 OrderID int NOT NULL,

 OrderNumber int NOT NULL,

 PersonID int,

 PRIMARY KEY (OrderID),

 FOREIGN KEY (PersonID) REFERENCES Person(PersonID));

Self Assignment

- Can other than Primary Key “PK” (Non-key Attributes) used a Foreign key ?
- How to drop a value of (Tuple) in parent table when it is been referred in child table (i.e. FK)?
- What is the minimum FK eligibility criteria for a column ?

Null Value Constraint

- Absence of any value (i.e., unknown or nonapplicable to a tuple)
- Example:
 - ❑ **CREATE** TABLE Persons (
 ID int NOT NULL,
 LastName varchar(255) NOT NULL,
 FirstName varchar(255) NOT NULL,
 Age int
);

Default Constraint

- Sets a default value for a column when no value is specified
 - ❑ CREATE TABLE Persons (
 ID int NOT NULL,
 LastName varchar(255) NOT NULL,
 FirstName varchar(255),
 Age int,
 City varchar(255) DEFAULT 'Sandnes'
);

Check Constraint

- CHECK constraint on the "Age" column when the "Persons" table is created. The CHECK constraint ensures that the age of a person must be 18, or older:
 - ❑ CREATE TABLE Persons (
 ID int NOT NULL,
 LastName varchar(255) NOT NULL,
 FirstName varchar(255),
 Age int,
 CHECK (Age>=18)
);

Schema and Subschemas



Internal Level
Physical Database



DBMS DBMS Software

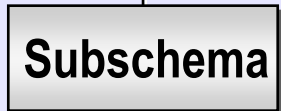
Some end-user applications can
be supported by *views*


Schema **Conceptual Level**

**External
Level**


Subschema


Subschema


Subschema


User

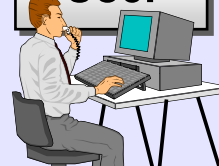
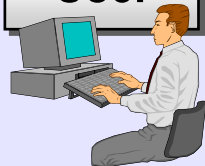

User


User


User


User


User



In Next Lecture

- Views

- Types of views

- Indexes

- SQL Indexes

Thanks