Introduction to Database Management Systems [CSCI 11062]

Relational Data Modeling and Relational Database Constraints

Dr. (Mrs.) Muditha Tissera
Email: muditha t@kln.ac.lk

Session Outcomes

LO2: Design a relational database system using the ER and EER Model

1. Introduce Relational Model
2. Introduce and formalize the Relation
3. Introduce Key Fields
4. Describe the Relational Model Constraints/Integrity Constraints
5. Work with Database operations with enforced Integrity Constraints

Relational Model

- The model was first proposed by Dr. E.F. Codd of IBM in 1970 in the following paper:
  "A Relational Model for Large Shared Data Banks," Communications of the ACM, June 1970.

  The above paper caused a major revolution in the field of Database management and earned Ted Codd the coveted ACM Turing Award.

- Everything is a relation (= table)

  Every relation is a table with rows & columns

  A Relation is a mathematical concept based on the ideas of set theory predicate logic.

- There are standard ways to convert from

  the E-R model (**conceptual model**)

  ⬇ ⬇

  Relational model (**logical model**)

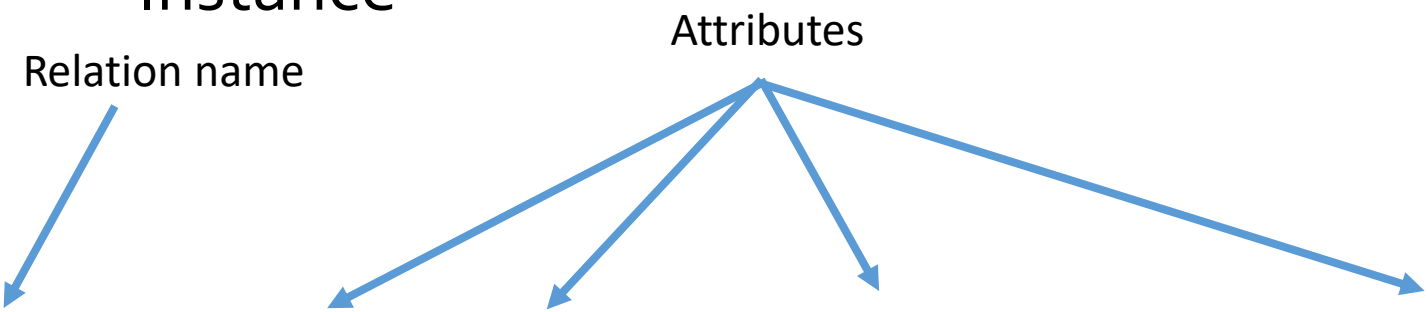
Relational Model (contd.)

- The relational model represents the database as a collection of **relations.**

- Relation consists of
  - Relation schema
  - Relation instance

Relation

- RELATION:
  - Schema
  - Instance



Customer

Tuples

Cust-id	Cust-name	Address	PhoneNo
124235	Kasun	43, Main Street, Matara	071 234 3451
312311	Amal	56, Gamunu Mawatha, Yakkala	077 456 4561
542341	Sumudu	12, Pasal Mawatha, Bambalapiytiya	null
967854	Amal	23, Kandy Road, Kiribathgoda	077 564 7651

Relation Schema

Describes the **structure** of the relation.

- name of the relation,
- name of each filed,
- domain of each field

Domain : Allowable values for the attribute.

is described by domain name and set of associated values

- E.g., “Sri Lanka phone numbers” are the set of 10 digit phone numbers
- domain of Cust-id is 6 digit numbers
- Dates have various formats such as month name, date, year or yyyy-mm-dd, or dd mm yyyy etc.

Relation Schema

Relation schema *R* is defined over **attributes** A1, A2, ...An with domains D1, D2, ….., Dn

The **Schema** of a Relation R: ***R* (A1, A2, ..., An)**

For Example -

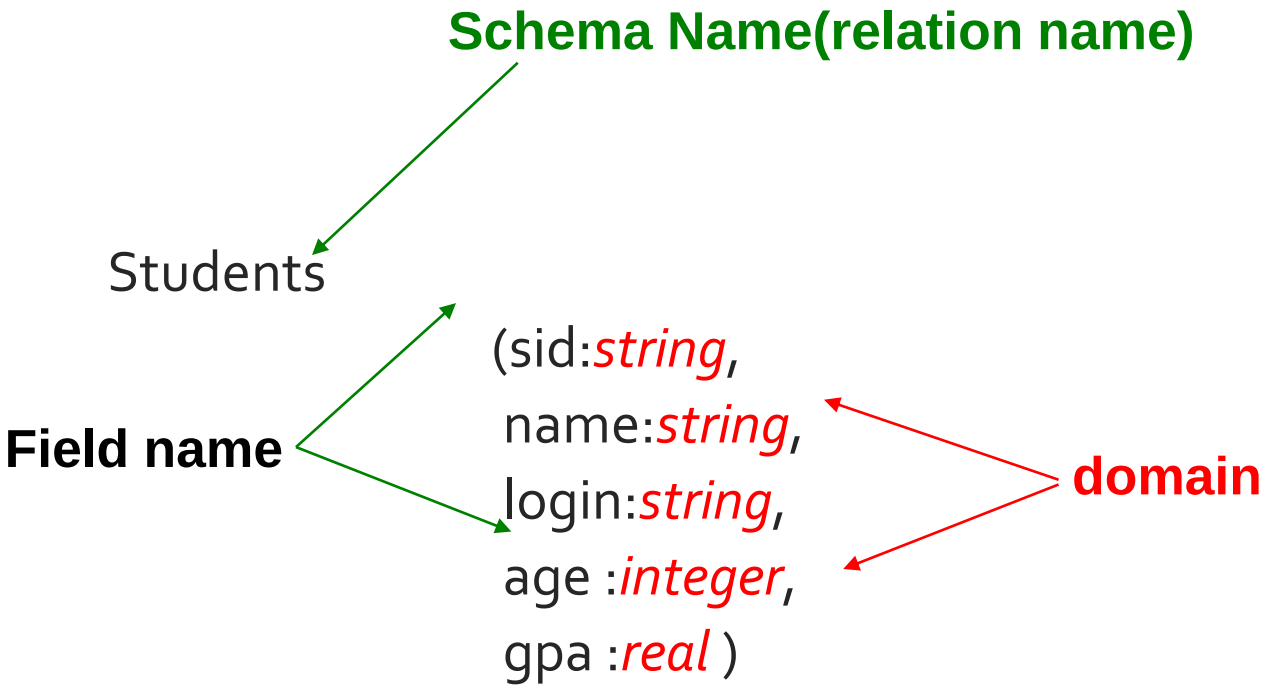
CUSTOMER (Cust-id, Cust-name, Address, PhoneNo)

Here, CUSTOMER is a relation defined over the four attributes Cust-id, Cust-name, Address, PhoneNo, each of which has a **domain** or a set of valid values. For example, the domain of Cust-id is 6-digit numbers.

Example-

CUSTOMER (Cust-id: Integer, Cust_name:String, Address:String, PhoneNo: String)

Relation Schema

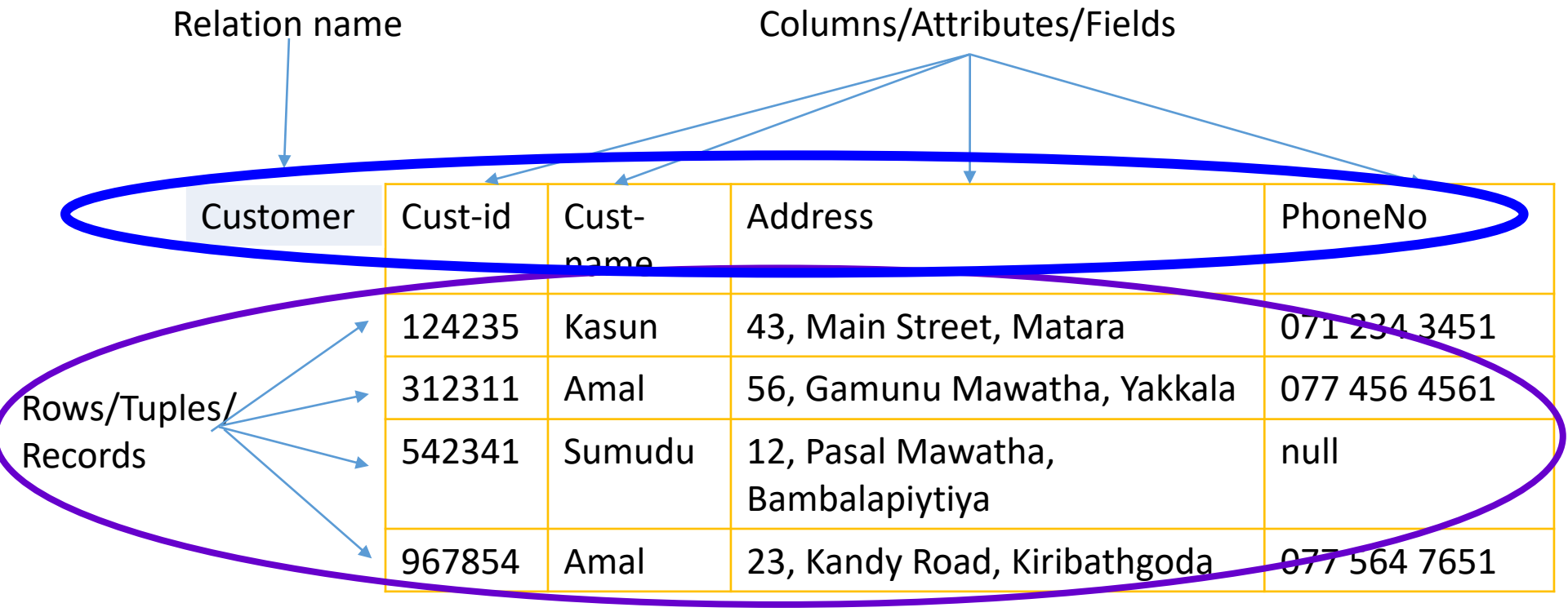


Domain GPA : real (0-4)

Relation Instance

- Set of tuples or records or rows :
- Each tuple has the same number of fields as the relation schema

  Example : Relation Instance



Customer	Cust-id	Cust-name	Address	PhoneNo
	124235	Kasun	43, Main Street, Matara	071 234 3451
	312311	Amal	56, Gamunu Mawatha, Yakkala	077 456 4561
	542341	Sumudu	12, Pasal Mawatha, Bambalapiytiya	null
	967854	Amal	23, Kandy Road, Kiribathgoda	077 564 7651

Degree (or Arity) of a relation

The **degree** of the relation R is the number of attributes in R.

The **cardinality** of the relation R is the number of tuples it contains.

ID	Name	Phone
S01	Mike	111
S02	Elisa	222

Degree = 4 (ID,Name,Address,Phone)

Cardinality = 2

This is the basic structure of the relation model, a table or relation.

Informal definition

Relation: A table of values

- A relation may be thought of as a set of rows.
- A relation may alternately be thought of as a set of columns.
- Each row represents a fact that corresponds to a real-world entity or relationship.
- Each row has a value of an item or set of items that uniquely identifies that row in the table.
- Sometimes row-ids or sequential numbers are assigned to identify the rows in the table.
- Each column typically is called by its column name or column header or attribute name.

Formalizing : Relations

Definition: A relation is a named table of data

Table is made up of rows (records or tuples), and columns (attributes or fields)

Not all tables qualify as relations. Requirements:

1. The relation has a unique name (that is distinct from all other names in the relational database schema).
2. Every attribute value is atomic (not multivalued, not composite)

   NULL is a special value to represent "**unknown**" or "**inapplicable**".

1. The values of an attribute are all from the same domain
2. Every row is unique (can't have two rows with exactly the same values for all their fields)
3. Attributes (columns) in tables have unique names
4. The order of the columns is irrelevant
5. The order of the rows is irrelevant

Informal Terms		Formal Terms
Table /File		Relation
Column/Field		Attribute/Domain
Row/Record		Tuple
Values in a column		Domain
Table Definition		Schema of a Relation
Populated Table		Extension/Instance/State

Introduce Key Fields

Primary key (PK)

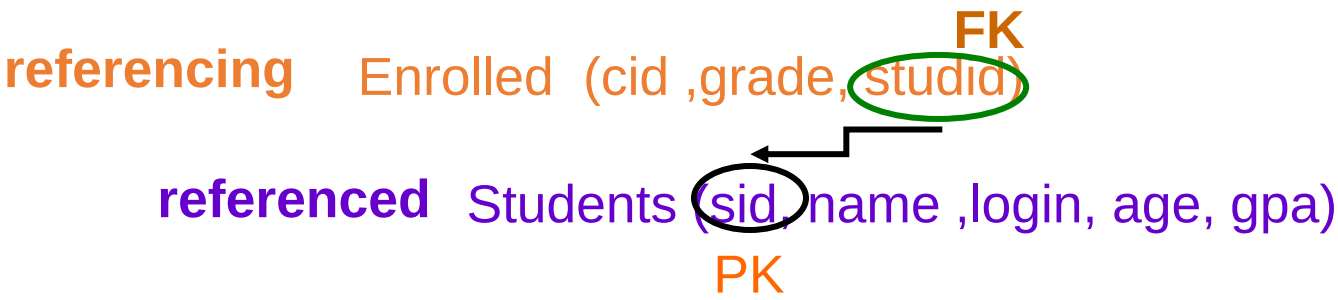
- Minimal set of attributes that uniquely identifies a row
- This is how we can guarantee that all rows are unique

Foreign key (FK)

Set of attributes in a table that serves as a reference to the primary key of another table

Foreign Key

- A constraint involving *two* relations
  - **referencing relation**
  - **referenced relation**.
- Tuples in the ***referencing relation*** have attributes **FK** (called **foreign key** attributes) that reference the primary key attributes **PK** of the ***referenced relation***



- Display the foreign keys by drawing an arrow from the **foreign key** to the **primary key**

Relational Model Constraints

Constraints on databases can generally be divided into three main categories:

1. **Inherent model-based constraint (implicit)**
   Constraints that are inherent in the data model.
   E.g., A relation cannot have duplicate tuples.

2. **Schema-based constraints (explicit)**
   Constraints that can be directly expressed in the schemas of the data model, by specifying them in the DDL.

3. **Application-based constraints (business rules/ semantic constraints)**
   Constraints that cannot be directly expressed in the schemas of the data model and be expressed by the application program.

Integrity Constraints [IC]

- DBMS must prevent entry of incorrect information.
- To prevent : Constraints / conditions are specified on a relational schema = Integrity Constraints (IC)
- Database which satisfies all constraints specified on a database schema is a legal instance(Valid state).
- DBMS enforces constraints - permits only legal instances to be stored
- When the application is run the DBMS checks for the violation and **disallows** the changes to the data that **violates the specified IC**

Integrity Constraints

- Integrity Constraints are **Specified** and **Enforced** at different times.
  - **Specified :** When the DBA /end user defines the data base schema
  - **Enforced :** When database application is run
    - DBMS checks for violations
    - Disallow violating entries

Integrity Constraints

Are of four types:

1. Domain constraints
2. Key constraints
3. Entity integrity constraints
4. Referential integrity constraints

Domain Constraints

- **Domain constraints**: value in the Column must be drawn from the domain associated with that column.

- **Restricts the :**
  - **Type**
  - **Values** that can appear in the field

Eg.

- Name **Char** (**25**)
- GPA (**real** >=**0**, =<**4**)

Disallow invalid values.

Key constraints

Is a statement that;

- A certain minimal subset of the fields of a relation is a unique identifier for a tuple.

Which Means

- Two tuples in a legal instance cannot have identical values in all the fields of a key.
- **All tuples in a relation must be distinct.**

Disallow duplicate key values

Constraints…

- **Entity Integrity Constraints:**

  Primary key values cannot be null

  This is because primary key values are used to *identify* the individual tuples.

Constraints…

- **Referential Integrity Constraints**
  - Sometimes, information stored in one relation is linked to information stored in another relation.
  - If one is modified the other must be modified to keep the data consistent.
  - An IC involving both relations must be specified.
  - IC involving 2 relations is a **foreign key constraint.**
  - Foreign keys enforce referential integrity constraints

Referential Integrity

- The value in the foreign key column (can be either:
  - a value of an existing primary key in the **referenced relation** or a **null**

Enrolled

cid	grade	stuid
Carnatic101	C	53666
Reggae203	B	53666
Topology112	A	53650
History105	B	53666

FK

PK

Students

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

Identify the Foreign Key/s, the referencing relation and referenced relation and draw the arc

S(SNO, SNAME, STATUS, CITY) **P(PNO, PNAME, PRICE, WEIGHT, CITY)**

J(JNO, JNAME, CITY)

SPJ(SNO, PNO, JNO, QTY)

DB operations & constraints

- ICs are specified when a relation is created and enforced (checked) when a relation is modified.

- 3 types of modifications (update operations) to the relation :
  - Insert : inserts a new tuple(s) into a relation.
  - Delete : delete tuple(s) in a relation.
  - Update : changes the values of some attributes in existing tuples .
- Integrity constraints should not be violated by the update operations.

Insert operation

The insert operation can violate the following constraints:

- Domain constraints (invalid value)
- Key constraints (duplicate key values)
- Entity integrity constraints (null primary key value)
- Referential integrity constraint (non-existing primary key value)

Examples - Domain constraints (Insert invalid value)

Students

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

Examples - Key constraints (Insert duplicate key values) ?

Students

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

Example - Entity integrity constraints (Insert null primary key value) ?

Students

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

Examples - Referential integrity constraint (Insert non-existing primary key value) ?

Students

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

Enrolled

cid	Grade	sid
IT	A	53666
IS	B	53650

Delete operation

- Delete operation can violate referential integrity.

Enrolled

cid	grade	stuid
Carnatic101	C	53666
Reggae203	B	53666
Topology112	A	53650
History105	B	53666

Students

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

- Two options:
  - Reject the deletion
  - Cascade the delete

Update operation

- Update operation can be considered as deleting a tuple and re-inserting the tuple with new values

- All constraints discussed in Insert & Delete need to be considered
  - Domain constraints (invalid value)
  - Key constraints (duplicate key values)
  - Entity integrity constraints (null primary key value)
  - Referential integrity constraint (non-existing primary key value)

In case of integrity violation, several actions can be taken:

- Cancel the operation that causes the violation (REJECT option)
- Perform the operation but inform the user of the violation
- Trigger additional updates so the violation is corrected (CASCADE option, SET NULL option)
- Execute a user-specified error-correction routine

Other Types of Constraints

Semantic Integrity Constraints: (General Constraints/ Business Rules)

- based on application semantics and cannot be expressed by the model.

- E.g., “the suppliers status cannot be greater than 50”
  “the weight of a part cannot be less than 4g”
  “Colombo suppliers salary should always be greater than 50,000”
  “ a suppliers salary cannot be greater than his managers salary”

- A *constraint specification language* may have to be used to express these

- SQL allows triggers and ASSERTIONS to allow for some of these

Consider the S, P, J and SPJ tables to answer these questions.

S

sno	sname	status	city	salary
S1	SAMAN	20	COLOMBO	45000
S2	JAGATH	10	KANDY	42000
S3	ANIL	30	KANDY	30000

SPJ

sno	pno	jno	qty
S1	P1	J1	200
S1	P1	J4	700
S1	P4	J1	800
S2	P1	J1	300
S2	P2	J4	700
S2	P3	J1	400
S2	P3	J2	200

If the following Insert operation is applied on the tables S, P and SPJ what are the constraints that are violated and specify any actions that could be used.

1. Insert <null, ‘Kamal’, 20, ‘Colombo’, 50000> into S
2. Insert <‘S2’, ‘Virul’, 10, ‘Kandy’, 35000> into S
3. Insert <999, ‘Anil’, 30, ‘Galle’, 25000> into S
4. Insert <‘S9’, ‘P1’, ‘J1’, 300 > into SPJ

Consider the S, P, J and SPJ tables to answer these questions.

S

sno	sname	status	city	salary
S1	SAMAN	20	COLOMBO	45000
S2	JAGATH	10	KANDY	42000
S3	ANIL	30	KANDY	30000

SPJ

sno	pno	jno	qty
S1	P1	J1	200
S1	P1	J4	700
S1	P4	J1	800
S2	P1	J1	300
S2	P2	J4	700
S2	P3	J1	400
S2	P3	J2	200

If the following delete operation is applied on the tables S and SPJ what are the constraints that are violated and specify any actions that could be used.

5. Delete the tuple with sno = ‘S1’ and pno = ‘P1’ from SPJ
6. Delete the tuple with sno = ‘S2’ from S

Relational Database Schema

Relational database:

A collection of normalized relations with distinct relation names.

Relational Database Schema (S):

A set of relation schemas that belong to the same database with distinct names

i.e. $S = \{R_1, R_2, ..., R_n\}$ and a set of integrity constraints IC

End of Lecture 05

Reference: Chapter 5

Ramez Elmasri and Shamkant B. Navathe (2017),
Fundamentals of Database Systems, , Addison-Wesley
Longman Publishing Co., 7th edition.

Questions ?