

The Relational Model (Part I)

Steps in Database Design

- Requirements Analysis
 - user needs; what must database do?
- Conceptual Design
 - high level description: ER
- **Logical Design**
 - translate ER into DBMS data model
- Schema Refinement
 - consistency, normalization
- Physical Design - indexes, disk layout
- Security Design - who accesses what, and how

Today's lecture

- **Introduction to DBMS data model**
 - Concepts (table, schema, row, column, ...)
 - Using SQL to create table, add and delete tuples
 - Integrity constraints

Relational Database: Definitions

- ***Relational database: a set of relations.***
- ***Relation: made up of 2 parts:***
 - *Schema* : specifies the name and *attributes* of relation
 - *Instance*
 - A *table* with rows and columns.
 - Consistent with schema

Relational Schema

- A *Schema* for a relation is represented by:
relation_name (attr1:type, ... attrn:type)
 - Example:
Students(*sid*: string, *name*: string, *login*: string, *age*: integer, *GPA*: real)
- Attributes are referenced by name, not column locations
- Column names must be unique

Exercise



Student	Course	Course
Anna	DB Mgt.	Operating System
Bob	DB Mgt.	Web Programming
Cathy	Operating System	Artificial Intelligence

- **Is this table good?**



Exercise (Cont.)

Student	Course1	Course 2
Anna	DB Mgt.	Operating System
Bob	DB Mgt.	Web Programming
Cathy	Operating System	Artificial Intelligence

- **Is this table good?**

Relational Instances

- Instance: a *table*, with rows and columns
- Attributes (or fields) are stored in columns.
- Tuples (or records) are stored in rows.
- Attributes have a domain – an atomic type.
- The cardinality of the relation R = the number of rows in R (excludes the first row!)
- The degree/arity of the relation R = the number of columns in R

Notes of Relational Model

- No duplicate tuples in a relation
 - What if we want to insert duplicate tuples?
What can we do?
- Ordering
 - No ordering of tuples in a relation
- The value of each attribute is either one drawn from its domain or the special value *NULL*
- Attribute's values are *atomic*.



Example Instance of Students Relation

Schema

Students(*sid*: string, *name*: string, *login*: string, *age*: integer, *GPA*: real)

SID	Name	Login	Age	GPA
53666	Jones	Jones@cs	18	3.4
53668	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

- Cardinality = 3, degree/arity = 5.

Questions:

- (1) Do the values in each column of a relation instance have to be distinct?
- (2) Does each tuple (i.e., each row) have to be distinct?

Exercises



- How many distinct tuples are there in a relation instance with cardinality 20?

Defining a Relation Schema in SQL

- **SQL (pronounced SEQUEL): standard language to describe and manipulate relational database**
 - Data Definition Language (DDL)
 - create, modify, delete relations
 - specify constraints
 - administer users, security, etc.
 - Data Manipulation Language (DML)
 - Specify *queries* to find tuples that satisfy criteria
 - add, modify, remove tuples

Creating Relations in SQL

- **SQL syntax**
 - CREATE TABLE <name> (<field> <domain>, ...)
- **Example: creates the Students relation.**

Schema

Students(*sid*: string, *name*: string, *login*: string, *age*: integer, *GPA*: real)

```
CREATE TABLE Students
(sid CHAR(20),
 name CHAR(20),
 login CHAR(10),
 age INTEGER,
 GPA FLOAT);
```

Note: the type
(domain) of each field
is specified, and
enforced by the DBMS

Creating Relations in SQL

- **Another example: the Enrolled table holds information about courses that students take.**

Schema

Enrolled (*sid*: string, *cid*: string, *grade*:real)

```
CREATE TABLE Enrolled
  (sid CHAR(20),
   cid CHAR(20),
   grade float);
```

Data Types

- All attributes must have a data type
 - 1) Character: strings of fixed or varying length
 - CHAR(n) → a fixed-length string of n characters
 - OR
 - VARCHAR (n)
 - 2) BOOLEAN: denotes an attribute whose value is logical
 - The possible values are TRUE, FALSE, and UNKNOWN
 - 3) DATE & TIME: represent dates and times

Data Types (Cont.)

- 4) INT or INTEGER: denotes typical integer values
- 5) FLOAT: denotes floating-point numbers
 - Real numbers with a fixed decimal point
 - DECIMAL (n,d) allows values that consists of n decimal digits, with the decimal point assumed to be *d* positions from the right

Case-sensitivity of Table/Column Names

- **Is the table/column name in SQL statement case sensitive?**
 - By default,
 - Case-sensitive on Linux
 - Case-insensitive on Windows
 - MySQL has a configuration option to enable/disable it.

Adding Tuples in SQL

- **SQL syntax**

- INSERT INTO <TableName> (<field names>)
VALUES(<field values>)

- **Insert a single tuple**

```
INSERT INTO Students (SID, Name, Login, Age, GPA)
VALUES('53688', 'Smith', 'smith@ee', 18, 3.2);
```

Adding Tuples in SQL (Cont.)

- **Insert multiple tuples**

SID	Name	Login	Age	GPA
53666	Jones	Jones@cs	18	3.4
53668	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

```
INSERT INTO Students (SID, Name, Login, Age, GPA)
VALUES ('53666', 'Jones', 'jones@cs', 18, 3.4),
      ('53690', 'Smith', 'smith@eecs', 18, 3.2),
      ('53650', 'Smith', 'smith@math', 19, 3.8);
```

Deleting Tuples in SQL

- **SQL syntax**

- DELETE FROM <TableName>
WHERE <condition>

- **Can delete all tuples satisfying some condition (e.g., name = Smith):**

```
DELETE FROM Students S  
WHERE S.name = 'Smith';
```

SID	Name	Login	Age	GPA
53666	Jones	Jones@cs	18	3.4
53668	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

Question: what will the table look like after deletion?

Integrity Constraints

- **Integrity constraints (ICs):** conditions specified on a database schema
- **Legal instances:** instances that satisfy ICs
- **Types of ICs**
 - Domain constraints: (e.g., age of students must be at least 18)
 - Keys
 - Foreign keys

Keys

- Keys: the minimal subset of the fields that uniquely identifies a tuple
- No part of a key can be null

SID	CID	Grade
53666	CS442	C
53666	CS105	B
53650	CS443	A
53666	CS510	B

Enrolled

SID	Name	Login	Age	GPA
53666	Jones	Jones@cs	18	3.4
53668	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

Students

Primary Keys

- **A set of fields is a superkey if:**
 - No two distinct tuples can have same values in all key fields
 - *Question:* all attributes of a relation together form a super key for the relation (true/false?)
- **A set of fields is a key for a relation if :**
 - It is a superkey
 - No subset of the fields is a superkey
 - i.e., a minimal superkey
- **>1 key for a relation?**
 - One of the keys is chosen (by DBA) to be the *primary key*.
 - The primary key of a relation cannot contain a NULL value as a value for their components

Exercise

SID	Name	Login	Age	GPA
53666	Jones	Jones@cs	18	3.4
53668	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

- **E.x.**
 - For the relation above, assume *sid* is a key for Students.
 - What about *name*? Is it a key?

- **Are the following keys correct?**

CID	Grade
CS442	C
CS105	B
CS443	A
CS510	B

Name	Grade
Alice.S	C
Alan.B	B
Adam.H	A
Alan.B	B

Key1: cid
Key2: {cid, grade}

Key1: student
Key2: {student, grade}