CSC 365-07: Introduction to Database Systems **Spring 2023**

Lecture 1: Database and DBMS

Instructor: Alex Dekhtyar *Ishaan Sathaye*

Introduction

Definition of a database and DBMS in Professor Notes.

CSC 365-07: Introduction to Database Systems **Spring 2023**

Lecture 2: Relational Data Model

Instructor: Alex Dekhtyar *Ishaan Sathaye*

Relational Data Model

Definition 1 *Relational data model is an approach to organizing collections of data*

- Relation
  - Relational Table ⟶ **Name + Schema**
    - Schema: List of attribute name + attribute type pairs
- Relational Database ⟶ **Collection of Relations tables**
- **Table Instance**: set of records with instantiated values of the attributes
  - Finite
  - Records, rows, tuples

One unit of data is called a **datum**.

Object, entity, event: description of one object, entity, event

- **Records** consist of attributes or fields (rows in the table).
- **Attributes** is a named container for a value of a specific type.

Database Table Constraint

Definition 2 *Limitations of table instances*

- **Candidate Key**: set or lists of attributes that uniquely define a record in a table, **minimal such set of attributes**, made up of multiple attributes sometimes.
  - **Every attribute is necessary.**

Examples

CSC 365 Example

Course Object:

- Prefix: CSC $\longrightarrow$ **String**
- Course #: 365 $\longrightarrow$ **Integer**
- Name: Introduction to Database Systems $\longrightarrow$ **String**
- Description: Basic Principles, ... $\longrightarrow$ **String**
- Units: 4 $\longrightarrow$ **Integer**

Department Object:

- Name: Computer Science and Software Engineering
- Abbreviation: CSSE
- Building: 14
- Room: 245
- College: CENG

Stringing these objects together based on relationship would make a **network model**.

Schema Example

```
Course(Prefix String, Course# Integer, Name String, Description
String, Units Integer)
```

Prefix	Course#	Name	Description	Units
CSC	365	Introduction to Database Systems	Basic Principles, ...	4
CSC	357	Systems Programming	...	4

```
Department(Name, College, Building, Room): Department would also have a table as well.
```

CSC 365-07: Introduction to Database Systems **Spring 2023**

Lecture 3: RDM Cont.

Instructor: Alex Dekhtyar *Ishaan Sathaye*

Relational Data Model

What makes a record unique?

- **Superkey**: any set of attributes that uniquely defines a record in a table
- **Primary Key**: candidate key chosen by you

CSC 365-07: Introduction to Database Systems **Spring 2023**

Lecture 4: SQL DDL and DML

Instructor: Alex Dekhtyar *Ishaan Sathaye*

MySQL Access

1. Server Address = host: **mysql.labthreesixfive.com**
2. Port: 3306
3. username
4. password

MySQL Database

- Namespace
- Collection of Tables
- Set of Permissions

Case Sensitivity

Case Sensitive

- Table Names
- Database Names

Not Case Sensitive

- Attribute Names
- SQL Keywords

Types

- **Numeric Types**
  - **Integer Types**
    - TINYINT
    - SMALLINT
    - MEDIUMINT

    - **INT**
    - BIGINT
  - **Floating Point Types**
    - **FLOAT**
    - **DOUBLE(P, D)**
    - **DECIMAL**

- **String Types**
  - **Character Types**
    - **CHAR(N)** $\longrightarrow$ **Fixed Length**
    - **VARCHAR(N)** $\longrightarrow$ **Variable Length**
    - TINYTEXT
    - **TEXT** $\longrightarrow$ for storing large amounts of text
    - MEDIUMTEXT
    - LONGTEXT

- **Date and Time Types**
  - **Date Types**
    - DATE
    - DATETIME
    - TIMESTAMP
    - TIME
    - YEAR

Data Definition Language (DDL)

Commands from DDL act upon the schema

- CREATE TABLE
- DROP TABLE
- ALTER TABLE

Define a Relational Table

Aspects needed to define a table:

- Table Name
- Attributes: Name + Type
- Constraints

```
CREATE TABLE <table_name> (
    <attribute_name> <sql_type> [<single_line_constraints>],
    ...,
    <attribute_name> <sql_type> [<single_line_constraints>] [,
    <constraints>[,
    <constraints>]
]);
```

Data Manipulation Language (DML)

Commands from DML act upon the instance.

- INSERT
- DELETE
- UPDATE

Inserting Data

```
INSERT INTO <table_name>(<attribute_name>, ...)
    VALUES (<value>, ...);
```

Supply values in order of attribute declarations in CREATE TABLE statement. Can omit the attribute names if values supplied are in the same order. If need to omit a value then omit that attribute name as well.

More on Constraints

- **[NOT] NULL** - attribute cannot be null
- **UNIQUE**
- **PRIMARY KEY**
- **FOREIGN KEY**
- **DEFAULT <exp>** - default value for attribute
- **AUTO_INCREMENT** - means that the attribute is an integer and is automatically incremented

Lab 2

MySQL Server

- LabThreeSixFive.com

- mysql command line client
- IDE (DatGrip)
- mysql connectivity from Python

Lab 2 uses Create Table, Drop Table, and Insert.

Code from Lab

```
show tables

CREATE TABLE Departments (

    DeptId INT PRIMARY KEY,
    Abbr VARCHAR(20) UNIQUE, -- UNIQUE makes candidate key
    Name VARCHAR(128) UNIQUE,
    College CHAR(10),
    Building INT,
    Room CHAR(6),
    -- set multiple candidate keys at the bottom
    UNIQUE(Building, Room),
    -- foreign key always a separate line statement:
    -- FOREIGN KEY(College) REFERENCES colleges(abbr)

);

describe colleges;
SELECT * FROM colleges;

show CREATE TABLE colleges;

show CREATE TABLE Departments;

INSERT INTO Departments
    VALUES(1, 'CSSE', 'Computer Science and Software Engineering', 'CENG', 14, '245');

INSERT INTO Departments(DeptId, Abbr, Name, College, Building, Room)
    VALUES(1, 'CSSE', 'Computer Science and Software Engineering', 'CENG', 14, '245');
```

CSC 365-07: Introduction to Database Systems **Spring 2023**

Lecture 5: DDL and DML Continued

Instructor: Alex Dekhtyar *Ishaan Sathaye*

DML

Updating Data

```
UPDATE <table_name>
    SET <attribute_name> = <value>
    WHERE <condition>;
```

Example

```
UPDATE colleges
    SET abbr = 'COSAM'
    WHERE abbr = 'COASM'
```

WHERE clause is a filter that determines which rows are updated.

Deleting Data

```
DELETE FROM <table_name> -- just this is a valid command to delete all rows
    WHERE <condition>;
```

DDL

Altering Tables

```
ALTER TABLE <table_name>
    <Command> <parameters>;
```

Commands

- ADD - add a column/attribute/key
- DROP
- MODIFY
- RENAME

Parameters

- COLUMN
- CONSTRAINT
- FOREIGN KEY
- PRIMARY KEY
- UNIQUE

Adding an attribute, dropping/adding a constraint, renaming a table, disable/enabling keys, and modifying attributes examples are in this professor notes: `4-SQLDDLDML.pdf`

CSC 365-07: Introduction to Database Systems **Spring 2023**

Lecture 6: DML/DDL Cont., WHERE Clause., and MySQL Conn.

Instructor: Alex Dekhtyar *Ishaan Sathaye*

Announcements

Running Scripts for Lab 2

Can run from command line using mysql command or using mysql client. For running using mysql command need to specify the database if not using default database.

```
source script.sql
```

DML/DDL

Data Manipulation works on instance and Data Definition works on schema.

Altering a Table

Modifying the schema. ALTER examples in class.

For CREATE TABLE, you can name constraints:

```
CREATE TABLE Example (
    Id int PRIMARY KEY,
    X INT,
    Y INT,
    CONSTRAINT Point UNIQUE (X, Y)
);
```

Updating and Deleting from Table: WHERE Clause

Ex. Deleting in Table

```
DELETE FROM test02
    WHERE b > c
```

This deletes rows where b is greater than c.

Ex. Deleting with Scope

```
DELETE FROM test02
```

```
FOR EACH ROW in test01
DO
    DELETE FROM test02 -- delete(row, condition)
    WHERE b > c
```

SQL Boolean Expressions

- 0, 1
- Builtin: IN(...) $\longrightarrow$ returns bool
- $< Expression > < op1 > < op2 >$ // can also use IN or LIKE
- $< Expression >$ AND $< Expression >$
- $< Expression >$ OR $< Expression >$
- NOT $< Expression >$

MySQL Connectivity

Breifly went over the Python examples on Course webpage that connect to MySQL server.

CSC 365-07: Introduction to Database Systems **Spring 2023**

Lecture 7: Python Connectivity and Relational Algebra

Instructor: Alex Dekhtyar *Ishaan Sathaye*

Python MySQL Connectivity

Relational Database is sitting on a server. It is listening for connections, and our program is a client that connects to the server via the port. Essentially, there is a pipe and a exchange of messages that is happening. Generates a connection object that stores info about how to properly access the database.

Package

```
import mysql.connector
```

Connection

5 Things Needed: Host, Port, Username, Password, Database (sometimes not necessarily)

These get passed to mysql.connector.connect() function. This returns a connection object. is_connected() returns a cursor object.

Cursor object that is returned from the connection object. Cursor object is used to execute queries.

Relational Algebra

Relational $\longrightarrow$ Database Model $\longrightarrow$ Relational Model. Algebra: set of elements & operations on elements

Relational Algebra is operations on relational tables.

Boolean Algebra introduces operations on truth values

- T, F
- $\tilde{}$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$

Notation

Upper case letters like R, S, T, R_1, S_7, ... are relational table names. Letters from first half of alphabet like A, B, C, ... are attributes names. R(A_1,..., A_n) are to represent schema. t, s, r $\in$ R are tuples. a_1, a_2, ... are values. Ex. t = (a_1,..., a_n) and it could be referred to as t.A_1 = a_1.

Operations

Binary

Unary

- Selection σ - filter rows
- Projection π - filter columns

Selection Operation

- $\sigma_{<selection condition>}(R)$ - returns rows that satisfy condition
- Selection Condition denoted by C
- Ex. C = $A_2 =' Riley' \wedge A_3 =' Hicks'$
- **Formal Notation:** $\sigma_C(R) = \{t \in R | t satisfies C\}$

Projection Operation

- $\pi_{<attribute list>}(R)$ - returns columns that are in attribute list
- F is the projection list which is a list of attributes
- Ex. F = $(B_1, ..., B_m)$ where $B_i \in A_1, ..., A_n$
- **Formal Notation:** $\pi_F(R) = \{t' | \exists, t \in R, s.t. \forall B \in F, t'.B = t.B\}$
- **Projection squeezes out duplicates**

CSC 365-07: Introduction to Database Systems **Spring 2023**

Lecture 8: Relational Algebra

Instructor: Alex Dekhtyar *Ishaan Sathaye*

Relational Algebra

R, S, T are relational tables and these are sets. Set Operations include Union, Intersection, Difference, Symmetric Difference...

Example:

Table 1: R

A	B	C
1	2	a
2	4	b
3	1	d
4	4	d
4	5	a

Table 2: S

B	D	E
2	a	1
3	a	2
7	b	3
5	b	1
1	c	2

Selection Review

Duplicate Elimination: $\sigma_{A=4 \vee B=4}(R)$

A	B	C
2	4	b
4	4	d
4	5	a

Projection Review

• Subset and Columns: $\pi_{B,D}(R)$. The schema here is $R(B, D)$.

• Can make composition of operations: $\pi_{B,E}(\sigma_{D=b}(R))$ because an result of an operation is a relational table. Result:

B	E
7	3
5	1

• Can also reorder columns: $\pi_{EBD}(R)$

• Duplicate Columns: $\pi_{ABA}(R)$. Issue with this is that there are 2 columns with the same name. Disambiguate by renaming: $\pi_{A_1 B A_2}(R)$

• Can also introduce new columns: $\pi_{A,B,2 \cdot A}(R)$

Cartesian Product

• $R \times S$ is the cartesian product of R and S. Result is a table with all possible combinations of rows from R and S.

• Notation: $R \times S = \{(t, t`)|t \in R, t \in S\}$

Ex. $\sigma_{A<3}(R) \times \sigma_{B<3}(S)$

A	B	C	B	D	E
1	2	a	2	a	1
1	2	a	3	a	2
1	2	a	5	b	1
2	4	b	2	a	1
2	4	b	3	a	2
2	4	b	5	b	1

Join

Table 3: R

Id	Customer	C
1	2	a
2	4	b
3	1	d
4	4	d
4	5	a

Table 4: S

Id	Name	E
2	a	1
3	a	2
7	b	3
5	b	1
1	c	2

• Who purchased Receipt 3 $\longrightarrow$ Find a person with Id 1: $\pi_{Name}(\pi_{R.Cust=S.Id}(\sigma_{Id=3}(R)) \times S)$

• Notation: $R \bowtie_C S = \sigma_{R.xoperationS.Y}(R \times S)$

CSC 365-07: Introduction to Database Systems **Spring 2023**

Lecture 9: Joins

Instructor: Alex Dekhtyar *Ishaan Sathaye*

Announcements

• Double-sided cheat sheet allowed for Midterm. • Homework assigned on Thursday as prep for Midterm.

Example

Table 5: T

Id	Name	Course	Grade
1	Joe	CSC 365	A
2	Mary	CSC 365	A
3	Lisa	CSC 101	B
4	Luis	STAT 301	C
5	Steve	STAT 301	A
1	Joe	CSC 357	B
2	Mary	CSC 357	C

Table 6: C

CourseId	Instructor
CSC 357	Nico
CSC 365	Migler
STAT 301	Bodwin
CSC 101	Rivera
STAT 302	Frame

Transcript Table T has primary key $(Id, Course)$ and foreign key $(Course)$ referencing Course Table C.

• **Find all Joe's instructors.**

$$\sigma_{Name=Joe}(T)$$
$$\pi_{Course}(\sigma_{Name=Joe}(T))$$
$$C \times \pi_{Course}(\sigma_{Name=Joe}(T))$$
$$\sigma_{C.CourseNo=T.Course}(C \times \pi_{Course}(\sigma_{Name=Joe}(T)))$$
$$\pi_{Instructor}(\sigma_{C.CourseNo=T.Course}(C \times \pi_{Course}(\sigma_{Name=Joe}(T))))$$

Cartesian Product in line 3 makes table with Course No., Instructor, and Course combination for Joe. So it would be every combination with Course and CourseNo.:

C.CourseNo	C.Instructor	T.Course
CSC 357	Nico	CSC 365
CSC 365	Migler	CSC 365
...	...	...
CSC 357	Nico	CSC 357
CSC 365	Migler	CSC 357
...	...	...

In Line 4, we are selecting the rows where the CourseNo. in C is equal to the Courses in T.

- **For each course grade, report the instructor who assigned it.**

$$\pi_{T.Course,T.Grade,C.Instructor}(\sigma_{T.Course=C.CourseId}(T \times C))$$

First, we are taking the Cartesian product which gives us 35 rows. Then we are making the table smaller by only selecting the rows that are equal in Courses.

Θ−Join

$$R \bowtie_{\Theta} S = \sigma_{\Theta}(R \times S)$$

Θ - selection condition where each comparison uses attributes from R and S

Equi-Join

$$R \bowtie_{\Theta} S$$

$$\Theta = (\text{R.A} = \text{S.B}) \wedge (\text{R.C} = \text{S.D}) \wedge (\text{R.E} = \text{S.F})$$

Natural Join

$$R(A_1, ...A_k, B_1, ...B_k)$$
$$S(B_1, ...B_k, C_1, ...C_k)$$

$$R \bowtie S = \pi_{R.*,S.C_1,...S.C_k}(R \bowtie_{Condition} S)$$

Condition is where $R.B_1 = S.B_1 \wedge ... \wedge R.B_k = S.B_k$.

Natural join looks at all **common attributes** of two relations and joins on them, removing one set of common attributes from the final relation.

Semi-Join

Special case of natural join where only attributes of one relation are kept. Essentially, it is a projection on all elements of 1 relation.

Left Semi Join:

$$R \bowtie_{\Theta} S = \pi_{R.*}(R \bowtie_{\Theta} S)$$

Right Semi Join:

$$R \bowtie_{\Theta} S = \pi_{S.*}(R \bowtie_{\Theta} S)$$

Example:

$$(\sigma_{Name=Joe}(R) \bowtie C)$$