

8. Relational Model and SQL

COMP1531 • KC Notes

8.1 Databases

- **Database**: logically coherent collection of related data
- **Database Management System (DBMS)**: application to:
 - Create and maintain a database (DDL)
 - Define **queries** that retrieves data
 - Perform **transactions** that modifies or deletes data (DML)
- **Data model**: how the data is structured in the database
 - **Relational model**: data is stored as a **set of records known as tables**
 - **Document model**: stored in a hierarchical fashion, e.g. XML
 - **Object-oriented model**: stored as a collection of objects
 - **Object-relational model**: hybrid model that combines relational and OO model
- **Relational Database Management System**
 - Stores data as **tuples/records in tables**
 - Allows user to **create relationships between tables**
- **Database schema**: adheres to a data model, **provides a logical view of the database**, defines how the data is organised and the relationships between them
- **Database schema instance**: state of database at a particular time

8.2 Relational Data Model

- Data is a **collection of interconnected relations (tables)**
 - Tables have **rows** of information, each row has **fields** known as **columns**

Mathematical	relation	tuple	attribute
Data-oriented	table	record (row)	field (column)

- **Relation** (table) (denoted R, S, T): a table, has a **name** and a **set of attributes**
 - Is a set of **tuples** with no ordering
 - Each relation has a **primary key** that is **unique over relation**
 - **STUDENT** table with attributes **name, zID, major**
- **Attribute** (denoted A, B or a_1, a_2): has a **name** and associated **domain** (set of allowed values)
 - **zID** attribute must be an **integer**
- **Tuple** (row): set of **values** which are:
 - **Atomic**: no multi-valued or composite attributes
 - **Belong to a domain** with a **name, data-type, format** with a common value NULL

STUDENT		
name	zID (Primary key)	major
Bob	5049392	Computer Science
Peter	5050291	Mechatronics

- **Database:** a set of relations

8.3 Relation Model Mathematical Representation

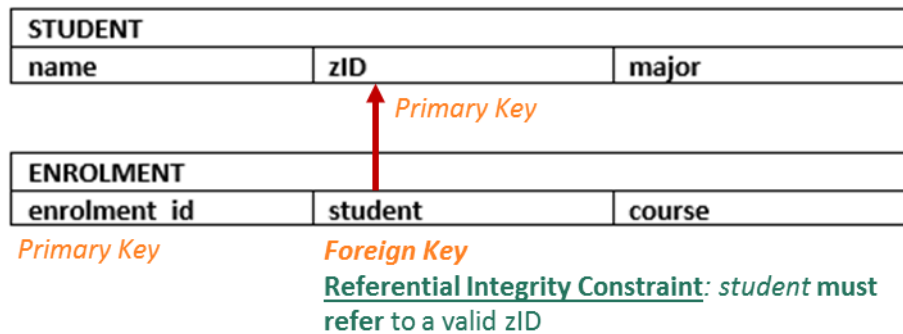
Mathematically:

Given a **relation** R_1 with n **attributes** a_1, a_2, \dots, a_n and corresponding **domains** $D_1, D_2 \dots D_n$:

- 1. Relation schema of R** $R_1(a_1: D_1, a_2: D_2, \dots, a_n: D_n)$
e.g. Student(name: string, zID: integer, major: string)
 - 2. Tuple of R** an element of $D_1 \times D_2 \times \dots \times D_n$ (i.e. a list of values)
e.g. ("Bob", 5049392, "Computer Science")
 - 3. Instance of R** a subset of $D_1 \times D_2 \times \dots \times D_n$ (i.e. a set of tuples)
e.g. { ("Bob", 5049392, "Computer Science"),
("Peter", 5050291, "Mechatronics") }
 - 4. Degree/Arity** number of attributes n of its relation schema
e.g. Student is a relationship of degree 7
-
- 5. Database schema** a collection of relation *schemas*
e.g. { R_1, R_2, \dots, R_n }
 - 6. Database (instance)** a collection of relation *instances*
e.g. { $r_1(R_1), r_2(R_2), \dots, r_n(R_n)$ }

8.4 Constraints

- **Domain constraints:** type or format constraints, NULL satisfies all domain constraints (except NOT NULL)
- **Key constraints:** whether a key needs to be unique (e.g. zID)
- **Entity integrity constraint:** whether an entity is well-defined or not (e.g. class must have a day, cannot be NULL)
- **Referential integrity constraints:** a set of attributes F in relation R_1 is a foreign key if
 - Attributes F **correspond to attributes in the primary key** of another relation R_2
 - The value of F in each tuple of R_1 **occurs as a primary key in R_2** or is entirely NULL



8.5 Making and changing a database

- To make a database:
 1. Choose a **data model**
 2. Set up structure by **defining a database schema** (e.g. for relational, define **tables, rows, columns, and types of fields, constraints, relationships**)
 3. Create initial state of the data
- When making **changes to relations**:
 - **Relation insert**: adding a new tuple (row) – easy
 - **Relation update**: changing attribute values – easy, you insert a new tuple into a set
 - **Schema update**: adding new attributes – difficult, you have to rewrite the whole file to modify each record

8.6 Relational Database Management Systems

- Relational model is a **mathematical theory**, relational DBMS **provides an implementation** of the model
- Features of RDBMS:
 - Supports **Large-scale data-intensive** applications
 - Provides efficient storage and retrieval (**disk/memory management**)
 - Supports multiple simultaneous **users (privilege, protection)** and multiple simultaneous **interactions (transactions, concurrency)**
 - Maintains reliable **access to stored data (backup, recovery)**
- **Structured Query Language (SQL)** – formalism to **express relational schemas**
 - Provides a **Data Definition Language (DDL)** for relations

```
CREATE TABLE company.employee (
  employee_id INT PRIMARY KEY,
  salary REAL NOT NULL CHECK (salary > 10000),
  department INT NOT NULL
);
```