**CSCI 4333.1 Classroom Notes and Demonstrations**

8/27/2025

**1.2 The Relational Model: an introduction**

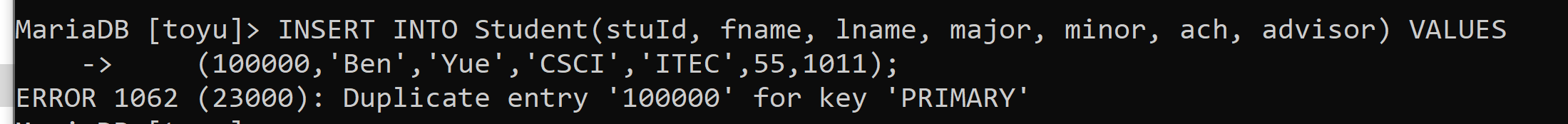
* The basic relational data model in layman terms:
  + A database is composed of a collection of *tables* (relations).
  + A table contains many *rows* (tuples) and *columns* (attributes/fields).
  + Each row contains many  *column values*.
  + Every row of a table has the same set of columns.
  + Values of the same column have the same data *type*.
  + Keys are sets of columns/attributes.
  + A *candidate key* of a table is a *minimal unique identifier* of a row in the table.
  + A *primary key* is a selected candidate key (for storing the table).

A screenshot of a computer

AI-generated content may be incorrect.

Table schema/structure: stuId is the primary key of the table student.

|  |
| --- |
| CREATE TABLE IF NOT EXISTS Student (  stuId INT NOT NULL,  fname VARCHAR(30) NOT NULL,  lname VARCHAR(30) NOT NULL,  major CHAR(4) NULL,  minor CHAR(4) NULL,  -- ach: accumulated credit hours, including transferred credits.  ach INTEGER(3) UNSIGNED NULL DEFAULT 0,  advisor INT NULL,  CONSTRAINT Student\_stuId\_pk PRIMARY KEY(stuId),  -- an artificial example of a CHECK constraint.  CONSTRAINT Student\_ach\_cc CHECK ((ach>=0) AND (ach < 250)),  CONSTRAINT Student\_major\_fk FOREIGN KEY (major)  REFERENCES Department(deptCode) ON DELETE CASCADE,  CONSTRAINT Student\_minor\_fk FOREIGN KEY (minor)  REFERENCES Department(deptCode) ON DELETE CASCADE,  CONSTRAINT Student\_advisor\_fk FOREIGN KEY (advisor)  REFERENCES Faculty(facId)  ); |



A screenshot of a computer

AI-generated content may be incorrect.

* + *Alternative/secondary keys* are candidate keys not selected as the primary key. (schoolName (i.e., {schoolName} is an alternative key of the table School in toyu)

Enroll table:

A screenshot of a computer

AI-generated content may be incorrect.

|  |
| --- |
| CREATE TABLE IF NOT EXISTS Enroll(  stuId INT NOT NULL,  classId INT NOT NULL,  grade VARCHAR(2) NULL,  n\_alerts INT NULL,  CONSTRAINT Enroll\_classId\_stuId\_pk PRIMARY KEY (classId, stuId),  CONSTRAINT Enroll\_classNumber\_fk FOREIGN KEY (classId)  REFERENCES Class(classId) ON DELETE CASCADE,  CONSTRAINT Enroll\_stuId\_fk FOREIGN KEY (stuId)  REFERENCES Student (stuId) ON DELETE CASCADE,  CONSTRAINT Enroll\_grade\_fk FOREIGN KEY (grade)  REFERENCES Grade (grade) ON DELETE CASCADE  ); |

* + A *foreign key* of a relation references a primary key of another relation (known as the parent or referenced table).

A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

|  |
| --- |
| CREATE TABLE IF NOT EXISTS Student (  stuId INT NOT NULL,  fname VARCHAR(30) NOT NULL,  lname VARCHAR(30) NOT NULL,  major CHAR(4) NULL,  minor CHAR(4) NULL,  -- ach: accumulated credit hours, including transferred credits.  ach INTEGER(3) UNSIGNED NULL DEFAULT 0,  advisor INT NULL,  CONSTRAINT Student\_stuId\_pk PRIMARY KEY(stuId),  -- an artificial example of a CHECK constraint.  CONSTRAINT Student\_ach\_cc CHECK ((ach>=0) AND (ach < 250)),  CONSTRAINT Student\_major\_fk FOREIGN KEY (major)  REFERENCES Department(deptCode) ON DELETE CASCADE,  CONSTRAINT Student\_minor\_fk FOREIGN KEY (minor)  REFERENCES Department(deptCode) ON DELETE CASCADE,  CONSTRAINT Student\_advisor\_fk FOREIGN KEY (advisor)  REFERENCES Faculty(facId)  ); |

More theoretically:

* The (theoretical) relational model is based on the concept of a relation.
* It is a *set-theoretic* model: the definitions are based on mathematical sets.
* If you are not familar with set theory, read about it. This is a basic, short, good, and good-enough introduction: <https://www.ucl.ac.uk/~ucahmto/0005_2021/Ch2.S1.html> (note that in the set builder form, the author used ":" to represent "such that". We usually use "|" instead.)
* Note that practical DBMS do not implement the pure relational model.
* In the theoretical relational model:
  1. An *attribute* (*column/field*) is a name.
  2. A *domain* is a *set* of values an attribute can take.
     1. It is the set of values of the*data type* of the attribute.
     2. The value of an attribute should be *atomic* (cannot be divided into smaller components with individual meanings):
        1. If all attributes of a relation are atomic, the relation is said to be in *First Normal Form*.
     3. *Null* may or may not be an acceptable value for an attribute. It depends on problem requirements.
  3. A *relation schema*, R, is a *set* of attributes A1, A2,…,An with their domains D1, D2,…, Dn.
  4. A *tuple* (*row*) is a *set* of *mapping* of a *set* of attributes to a *set* of values: Ai -> di where di ∈ Di, for i = 1 to n (∈: belongs to)
  5. A *relation* (*instance*) is a set of tuples.
  6. The *degree* (or *arity*) of a relation is the number of attributes in its schema.
* Some advantages of the *relational model* and relational DBMS (as compared to other databases):
  1. Strong mathematical foundation
  2. Simple
  3. Strong design theory
  4. Strong support of data integrity and consistency
  5. Strong support of transactions
  6. Strong industrial support and community
  7. High popularity
* Some disadvantages of the relational model and relational DBMS:
  1. The data model may not match the problem requirements well.
  2. Impedance mismatch with object-oriented models.
  3. Do not scale well.
  4. Structured data may be too restrictive for specific problems.