**Class Notes - CS482 - Chapter 2**

# 1 A Student Registration System

A project, that starts with a statement of the following objectives:

1. Authenticate themselves as users of the systems
2. Register and deregister for courses
3. Obtain report on a particular student's status
4. Maintain information about students and courses
5. Enter final grades for courses that a student has completed.

We will have the first specification similar to this for our project.

# 2 Relational Databases

Mainly interested in **relational model**.

**Relation and tuples.** Databases are stored in *tables*. Each row in a table represents an object or an entity of interest. Each column in the table describes the object in a particular way. Columns are *attributes* and each has its *domain*.

**Example 1** *The student registration system should contain a table about students with the attributes like Id, Name, Address, Status. Each row represents a student. Each column describes the student in a way.*
*The* current information *might look as follows:*

| *Id* | *Name* | *Address* | *Status* |
|---|---|---|---|
| *111111111* | *John Doe* | *123 Main St.* | *Freshman* |
| *666666666* | *Joseph Public* | *666 Hollow Rd.* | *Sophomore* |
| *111223344* | *Mary Smith* | *1 Lake St.* | *Freshman* |
| *987654321* | *Bart Simpson* | *Fox 5 TV* | *Senior* |
| *023456789* | *Homer Simpson* | *Fox 5 TV* | *Senior* |
| *123454321* | *Joe Blow* | *6 Yart Ct.* | *Junior* |

*The* STUDENT *Table*

*Relation.* A relation is a set of *tuples*. A relation could be viewed as a predicate with variables. The set of tuples defines the predicate (if a tuple belongs to the relation then it is true; otherwise, it is false)
*Tables vs. relation:* Each table is a relation. Each row is a tuple of the relation. For example, the first row of the above table can be viewed as the atom STUDENT(111111111, John Doe, 123 Main St., Freshman).

**Operations on tables.** Needed since we would like to search for and/or combine information from tables, which might be huge. E.g., searching for something in a table, joining different tables to create new one, updating certaint row, etc.
Several mathematical operations on relations can be used to define operations on tables.
Commercial DBMS helps make things simpler: it provides a convenient way for us to execute different operation on the tables; we only need to express what we want (as a *query*) and the DBMS will compute the result for us, i.e., we do not worry about *how can it be done*. Better DBMS is often equipped with more efficient engine to answer our queries.

**SQL.** A language that DBMS supports for users to query databases.

**SQL: Basic SELECT statement.** *Syntax:* `SELECT .... FROM .... WHERE ....`
*Example:*

```
SELECT Name FROM STUDENT WHERE Id = '111111111'
SELECT Id, Name FROM STUDENT WHERE Status = 'Senior'
SELECT * FROM STUDENT WHERE Status = 'Senior'
SELECT COUNT(*) FROM STUDENT WHERE Status = 'Senior'
```

**Multi-table SELECT statement.** Extended syntax: FROM's part includes a list of tables and WHERE's part is an expression over the attributes in the tables.
Suppose that grades are stored in a table named TRANSCRIPT with the attributes StudId, CrsCode, Semester, Grade.

```
SELECT Name, CrsCode, Grade
FROM STUDENT, TRANSCRIPT
WHERE StudId = Id and Status = 'Senior'
```

**Note:** See the difference in the condition 'WHERE'; the first equality combine the two tables in a certain way; the second one plays the role of a filter.

**Query optimization.** There are different ways for expressing a query. DBMS implements one way and often tries to optimize (the time) for computing the answers. However, we should learn to know how things can be done fast. Think about sorting!

**Chaning the content of a table.** SQL provides commands for updating, inserting, and deleting a rows, a set of rows, or a table.

```
UPDATE STUDENT SET Status = 'Sophomore' WHERE Id = '111111111'
INSERT INTO STUDENT(Id, Name, Address, Status)
VALUES('999000999', 'Athur Ni', '123 Hallowen', 'Freshman')
DELETE FROM STUDENT WHERE Id = '111111111'
```

**Creating tables and specifying constraints.** SQL allows us to create tables.

```
CREATE TABLE      STUDENT(
Id                INTEGER,
Name              CHAR(20),
Address           CHAR(50),
Status            CHAR(10),
PRIMARY KEY(Id))
```

# 3 ACID Properties of Transaction

Transactions are programs that interact with databases (through the DBMS) to maintain the *correctness* of the databases. It is easier to think about the ACID properties if you think about your bank account and the database (of bank accounts) that your bank has.

**Consistency.** Executing a transaction at a state in which the database is consistent will result in a consistent database.

*Withdrawing $100 from an ATM does not only reduce your account's balance but also reduces the total amount of cash available at the ATM. And, you cannot get the $100 if the balance is less than $100.*

**Atomicity.** The system must ensure that the execution of a transaction is completed as a whole or none.

*If you withdraw $100 and get the money, the balance must be reduced, the total amount of cash must be reduced. If you withdraw $100 and did not get the money, nothing should change.*

**Durability.** If a transaction is committed, its effects must remain throughout the lifetime of the database.

*If you withdraw $100, get the money, and your balance is 0, then it will stay so until you deposit some money.*

**Isolation.** The overall effect of the execution of several transactions must be the same as if they are executed sequentially.

*Withdrawing $100 on two ATM simultaneously does not make you richer!*

**Why ACID?** Database has consistency constraints or integrity constrains (IC); ICs are the characteristics of the information and must be maintaind **at all time**.

For example, each student has a unique Id; a student cannot take two different courses at the same time; grades are letters 'A', 'B', 'C', 'D', 'F', 'I', 'S', or 'U'; the GPA of a student is the average of his grades; etc.

Transactions must ensure that they do not leave the database in a state violating the ICs of the system. Violating one of the ACID properties will lead to the violation of the ICs. Thus the requirement!

# 4 Exercises

- **2.1**

  | | | |
  |---|---|---|
  | MARRIED(Husband, Wife) | = | $\{(a,b) \mid$ a is husband of b$\}$ |
  | BROTHER(Brother, Person) | = | $\{(c,d) \mid$ c is brother of d$\}$ |
  | SIBLING(Person_1, Person_2) | = | $\{(e,f) \mid$ e is a sibling of f$\}$ |
  | BROTHER_IN_LAW | = | $\{(c,b) \mid \exists a$ such that MARRIED(a,b) and BROTHER(c,a)$\}$ $\cup$ |
| | | $\{(c,a) \mid \exists b$ such that MARRIED(a,b) and BROTHER(c,b)$\}$ $\cup$ |
| | | $\{(a,c) \mid \exists b$ such that MARRIED(a,b) and SIBLING(c,b)$\}$ |

- **2.2**

  COURSESREGISTERFOR(Id, CrsCode)

  COURSESTAKEN(Id, CrsCode, Grade)

- **2.4**

```
CREATE TABLE  TRANSCRIPT(
StudId        INTEGER,
CrsCode       CHAR(6),
Semester      CHAR(6),
Grade         CHAR(1),
PRIMARY KEY   (StudId, CrsCode, Semester))
```