LA IMMACULADA CONCEPCION SCHOOL
SENIOR HIGH SCHOOL
GRADE 12 – ICT: COMPUTER PROGRAMMING

APRIL 27 – 30, 2020
Topic: Relational Data Model and Relational Database Constraints

Data models that preceded the relational Model include the hierarchical and network models. They were proposed in the 1960s and were implemented in early DBMSs during the late 1960s and early 1970s. Because of their historical importance and existing user base for these DBMS.

Relational Model Concepts
The relational model represents the database as a collection of *relations*. Informally, each relation resembles a table of values or, to some extent, a *flat* file of records. It is called a **flat file** because each record has a simple linear or flat structure. For example, the database of files is similar to the basic relational model representation. However, there are important differences between relations and files.

When a relation is thought of as a **table** of values, each row in the table represents a collection of related data values, a row represents a fact that typically corresponds to a real-world entity or relationship. Table name and column names are used to help to interpret the meaning of the values in each row.
For example, the first table is called STUDENT because each row represents facts about a particular student entity. The column names- Name, Student_Number, Class and Major-specify how to interpret the data values in each row, based on the column each value is in. all values in a column are of the same data type.

In the formal relational model terminology, a row is called a *tuple*, a column header is called an *attribute,* and the table is called a *relation.* The data type describing the types of values that can appear in each column is represented by a *domain* of possible values. We now define these terms – *domain, tuple, attribute* and *relation* – formally.

Domains, Attributes, Tuples and Relations
A **Domain** *D* is a set of atomic values. By **atomic** we mean that each value in the domain is indivisible as far as the formal relational model is concerned. A common method of specifying a domain is to specify a data type from which the data values forming the domain are drawn. It is also useful to specify a name for the domain, to help in interpreting its values. an example of domain:

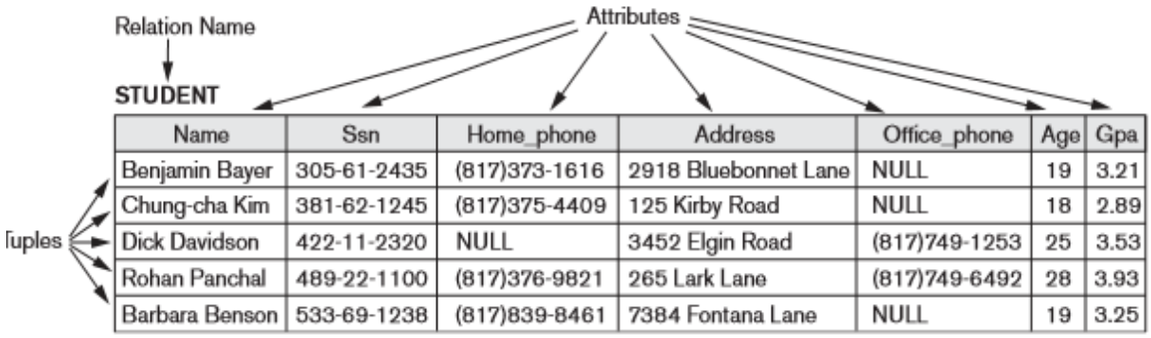
- Local_Phone_number. Set of seven-digit phone numbers valid within a particular area code in the US.

The preceding are called *logical* definitions of domains. A data type or format is also specified for each domain. For examples, the data type for the domain Usa_phone_numbers can be declared as a character string of the form (ddd)ddd-dddd, where each *d* is a numeric (decimal) digit and the first three digits form a valid telephone area code.

A relation schema *R*, denoted by *R(A1,A2,...,An)*,is made up of a relation name R and a list of attributes, *A1, A2, ...,An*. Each attribute Ai is the name of a role played by some domain D in the relation schema R. D is called the domain of Ai and is denoted by dom(Ai). A relation schema is used to describe a relation; R is called the name of this relation.
The degree (orarity) of a relation is the number of attributes n of its relation schema.

A relation of degree seven, which stores information about university students, would contain seven attributes describing each student.as follows: STUDENT(Name, Ssn, Home_phone, Address, Office_phone, Age, Gpa) Using the data type of each attribute,the definition is sometimes written as: STUDENT(Name: string, Ssn: string, Home_phone: string, Address: string, Office_phone: string, Age: integer, Gpa: real)
For this relation schema, STUDENT is the name of the relation, which has seven attributes. In the preceding definition, we showed assignment of generic types such as string or integer to the attributes. More precisely, we can specify the following previously defined domains for some of the attributes of the STUDENT relation: dom(Name) = Names; dom(Ssn) = Social_security_numbers; dom(HomePhone) = USA_phone_numbers3, dom(Office_phone) = USA_phone_numbers, and dom(Gpa) = Grade_point_averages. It is also possible to refer to attributes of a relation schema by their position within the relation; thus, the second attribute of the STUDENT relation is Ssn, whereas the fourth attribute is Address.

A relation (or relation state)4 r of the relation schema R(A1, A2, ..., An), also denoted by r(R),is a set of n-tuplesr= {t1,t2,...,tm}.Each n-tuplet is an ordered list of n values t =<v 1,v 2, ..., vn>, where each value vi, $1 \leq i \leq n$, is an element of dom (Ai) or is a special NULL value. (NULL values are discussed further below and in Section 3.1.2.) The ith value in tuple t, which corresponds to the attribute Ai, is referred to as t[Ai] or t.Ai (or t[i] if we use the positional notation). The terms relation intension for the schema R and relation extension for a relation state r(R) are also commonly used.



Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Benjamin Bayer	305-61-2435	(817)373-1616	2918 Bluebonnet Lane	NULL	19	3.21
Chung-cha Kim	381-62-1245	(817)375-4409	125 Kirby Road	NULL	18	2.89
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	(817)749-1253	25	3.53
Rohan Panchal	489-22-1100	(817)376-9821	265 Lark Lane	(817)749-6492	28	3.93
Barbara Benson	533-69-1238	(817)839-8461	7384 Fontana Lane	NULL	19	3.25

The attributes and tuples of a relation STUDENT.

To do:

1. Define the following terms as they apply to the relational model of data: *domain, attribute, n-tuple, relation schema, relation state, degree of a relation, relational database schema,* and *relational database state.*
2. Why are tuples in a relation not ordered?
3. Why are duplicate tuples not allowed in a relation?

Exercises

Suppose that each of the following Update operations is applied directly to the database state shown in the figure below. Discuss all integrity constraints violated by each operation, if any, and the different ways of enforcing these constraints.

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

LA IMMACULADA CONCEPCION SCHOOL
SENIOR HIGH SCHOOL
GRADE 12 – ICT: COMPUTER PROGRAMMING

Lesson 9: Relational and Relational Database Constraints

In a relational database there will typically be many relations, and the tuples in those relations are usually related in various ways. The state of the whole database will correspond to the state of all its relations at a particular point in time. There are generally many restrictions or **constraints** on the actual values in a database state. These constraints are derived from the rules in the mini world that the database represents.

Constraints on databases can generally be divided into three main categories:

1. Constraints that are inherent in the data model. We call these **inherent model-based constraints** or **implicit constraints.**
2. Constraints that can be directly expressed in schemas of the data model, typically by specifying them in the DDL (data definition language). We call these **schema-based constraints** or **explicit constraints.**
3. Constraints that *cannot* be directly expressed in the schemas of the data model, and hence must be expressed and enforced by the application programs. We call these **application-based** or **semantic constraints** or **business rules.**

The characteristics of relations are the inherent constraints of the relational model and belong to the first category. For example, the constraint that a relation cannot have duplicate tuples is an inherent constraint. The constraints we discuss in this section are of the second category, namely, constraints that can be expressed in the schema of the relational model via DDL.
Constraints in the third category are more general, relate to the meaning as well as behavior of attributes, and are difficult to express and enforce within the data model, so they usually checked within the application programs that perform database updates.

Another important category of constraints:
Data dependencies – Include functional dependencies and multivalued dependencies.
These are used mainly for testing the goodness of the design of a relational database and are utilized in a process called *normalization*.

Schema-based constraints include domain constraints, key constraints, constraints on NULLs, entity integrity constraints, and referential integrity constraints.

Domain Constraints

- Specifies that within each tuple, the value of each attribute *A* must be an atomic value from the domain dom. The data types associated with domains typically include standard numeric data types for integers (short integer, integer, and long integer) and real numbers (float and double precision float). Characters, Booleans, Fixed-length strings, and variable-length strings are also available, as are date, time, timestamp, and money or other special data types.

Key Constraints and Constraints on NULL Values

- A *relation* is defined as a *set of tuples*. All elements of a set are distinct. All tuples in a relation must also be distinct. This means that no two tuples can have the same combination of values for all their attributes.
- There are other **subsets of attributes** of a relation schema *R* with the property that no two tuples in any relation state *r* of *R* should have the same combination of values for these attributes. Suppose that we denote one such subset of attributes by SK; Then for any two distinct tuples t1 and t2 in a relation state *r* of *R,* we have the constraint.
- Any such attributes SK is called a **superkey** of the relation schema *R.* A superkey SK specifies a uniqueness constraint that no two distinct tuples in any state r of R can have the same value for SK. Every relation has at least one default superkey.
- A Superkey can have redundant attributes, however, so a more useful concept is that of a key, which has no redundancy.
- A Key of K of a relation schema R is a superkey of R with the additional property that removing any attribute A from K leaves a set of attributes K that is not a superkey of R any more.

A key satisfies two properties:

1. Two distinct tuples in any state of the relation cannot have identical values for (all) the attributes in the key. This first property also applies to a superkey.
2. It is a minimal superkey that is a superkey from which we cannot remove any attributes and still have the uniqueness constraint in condition 1 hold. So this property is not required by a superkey.

In general:
A relation schema may have more than one key, in this case, each of the keys is called a candidate key. It is also common to designate one of the candidate keys as the primary key of the relation.

Relational Databases and Relational Database Schemas
A relational database usually contains many relations, with tuples in relations that are related in various ways.

A **Relational database schema** *S* is a set of relation schemas S = {R1, R2,...,Rm} and a set of **integrity constraints** IC.
A database state that does not obey all the integrity constraints is called an **invalid state**.

Integrity, Referential Integrity, and Foreign Keys

Entity integrity constraint – states that no primary key value can be NULL. This is because the primary key value is used to identify individual tuples in two relations.

To define referential integrity more formally, first define the concept of a foreign key. The conditions for a foreign key specifies a referential integrity constraint between the two relation schemas of R1 and R2. A set of attributes FK in relation schema R1 is a foreign key of R1 the references relation R22 if it satisfies the following rules:

1. The attributes in FK have the same domain(s) as the primary key attributes PK of R2; the attributes PK of R2; the attributes.
2. A value of FK in a tuple t1 of the current state r1(R1) either occurs as a value of PK for some tuple t2 in the current state r2(R2) or is NULL. In the former case, we have t1[FK] = t2[PK], and we say that the tuple t1 references or refers to the tuple t2.

R1 is called the referencing relationandR2 is the referenced relation. If these two conditions hold, a referential integrity constraint from R1 to R2 is said to hold.

Update Operations, Transactions, and Dealing with Constraint Violations

The operations of the relational model can be categorized into retrievals and updates. The relational algebra operations, which can be used to specify retrievals. A relational algebra expression forms a new relation after applying a number of algebraic operators to an existing set of relations; its main use is for querying a database to retrieve information. The user formulates a query that specifies the data of interest, and a new relation is formed by applying relational operators to retrieve this data. That result relation becomes the answer to the user's query.

There are three basic operations that can change the states of relations in database.: Insert, Delete and Update (or Modify)
Insert – Used to insert one or more new tuples in a relation
Delete – used to delete tuples
Update (or Modify) – Used to change the values of some attributes in existing tuples.

The insert operation

- The Insert operation provides a list of attribute values for a new tuple t that is to be inserted into a relation R. Insert can violate any of the four types of constraints discussed in the previous section. Domain constraints can be violated if an attribute value is given that does not appear in the corresponding domain or is not of the appropriate data type.

The delete operation

- The Delete operation can violate only referential integrity. This occurs if the tuple being deleted is referenced by foreign keys from other tuples in the database. To specify deletion, a condition on the attributes of the relation selects the tuple (or tuples) to be deleted.

The update Operation

- The Update (or Modify) operation is used to change the values of one or more attributes in a tuple (or tuples) of some relation R. It is necessary to specify a condition on the attributes of the relation to select the tuple (or tuples) to be modified.

The Transaction Concept

- A database application program running against a relational database typically executes one or more transactions. A transaction is an executing program that includes some database operations, such as reading from the database, or applying insertions, deletions, or updates to the database. At the end of the transaction, it must leave the database in a valid or consistent state that satisfies all the constraints specified on the database schema.

LA IMMACULADA CONCEPCION SCHOOL
SENIOR HIGH SCHOOL
GRADE 12 – ICT: COMPUTER PROGRAMMING

Lesson 9: Relational and Relational Database Constraints
Student Task:

Directions: answer the following questions. Write your answer in a sheet of yellow paper.

1. What is the difference between a key and a superkey?
2. Why do we designate one of the candidate keys of a relation to be primary key?
3. Discuss the characteristics of relations that make them different from ordinary tables and files.
4. Discuss various reasons that lead to the occurrence of NULL values in relations.
5. Discuss entity integrity and referential integrity constraints. Why is each considered important?
6. Define foreign key. What is this concept used for?
7. What is a transaction? How does is differ from an Update Operation?