CSE 4409: Database Management Systems II

Abu Raihan Mostofa Kamal

Professor, CSE Department
Islamic University of Technology (IUT)
Gazipur-1704 Bangladesh
✉ raihan.kamal@iut-dhaka.edu
☎ +8801843925543

January 17, 2024

Chapter Outline

Course Outline

Pre-requisite: **CSE 4307** (Database Management Systems)

Syllabus: Part I (Revisit Database Basics)

Relational Database Programming: Introduction, its role in S/Wdevelopment; Relational Database Basic Constructs: Table, Keys, Views, Cardinality; Introduction to SQL, Relational query and sub- query, joins.

Tables (relation schema) and Keys

- It is a data-structure for storing records (in the secondary storage)

Tables (relation schema) and Keys

- It is a data-structure for storing records (in the secondary storage)
- It is user-defined

Tables (relation schema) and Keys

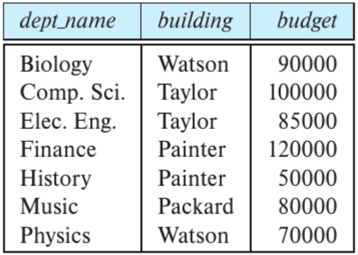
- It is a data-structure for storing records (in the secondary storage)
- It is user-defined
- It has a name and defined by a number of attributes and other constraints

Tables (relation schema) and Keys

- It is a data-structure for storing records (in the secondary storage)
- It is user-defined
- It has a name and defined by a number of attributes and other constraints

Tables (relation schema) and Keys

- It is a data-structure for storing records (in the secondary storage)
- It is user-defined
- It has a name and defined by a number of attributes and other constraints



dept_name	*building*	*budget*
Biology	Watson	90000
Comp. Sci.	Taylor	100000
Elec. Eng.	Taylor	85000
Finance	Painter	120000
History	Painter	50000
Music	Packard	80000
Physics	Watson	70000

Figure 1: Table

Keys

- We must have a way to specify how tuples (records) within a given relation (table) are **distinguished**.

Keys

- We must have a way to specify how tuples (records) within a given relation (table) are **distinguished**.
- The attribute values of a tuple must be such that they can **uniquely identify the tuple**. Otherwise record duplication will occur.

Keys

- We must have a way to specify how tuples (records) within a given relation (table) are **distinguished**.
- The attribute values of a tuple must be such that they can **uniquely identify the tuple**. Otherwise record duplication will occur.
- Super Key, Candidate Key, *Primary Key, Foreign Key*

Keys (Cont.1)

- A **superkey** is a set of one or more attributes that, taken collectively, allow us to identify uniquely a tuple in the relation. A superkey may contain **extraneous attributes**.

Keys (Cont.1)

- A **superkey** is a set of one or more attributes that, taken collectively, allow us to identify uniquely a tuple in the relation. A superkey may contain **extraneous attributes**.
- **Minimal superkeys** are called **candidate keys**.

Keys (Cont.1)

- A **superkey** is a set of one or more attributes that, taken collectively, allow us to identify uniquely a tuple in the relation. A superkey may contain **extraneous attributes**.
- **Minimal superkeys** are called **candidate keys**.
- We shall use the term **primary key** to denote a candidate key that is **chosen by the database designer** as the principal means of identifying tuples within a relation.

How to choose Primary Key

- It should be **informative** to the end user.

How to choose Primary Key

- It should be **informative** to the end user.
- It should be designed in a way that is **not changeable** in any case.

How to choose Primary Key

- It should be **informative** to the end user.
- It should be designed in a way that is **not changeable** in any case.
- A wise **trade-off** is desired.

Foreign Key

- A FOREIGN KEY is an attribute (or collection of attributes) in one table, that refers to the PRIMARY KEY in another table.

Foreign Key

- A FOREIGN KEY is an attribute (or collection of attributes) in one table, that refers to the PRIMARY KEY in another table.
- It provides a **link** between data in two tables.

Foreign Key

- A FOREIGN KEY is an attribute (or collection of attributes) in one table, that refers to the PRIMARY KEY in another table.
- It provides a **link** between data in two tables.
- It **removes redundancy and inconsistency**. (How?)

Foreign Key

- A FOREIGN KEY is an attribute (or collection of attributes) in one table, that refers to the PRIMARY KEY in another table.
- It provides a **link** between data in two tables.
- It **removes redundancy and inconsistency**. (How?)
- It ensures that a data would come from a **specific source**.

Views: A Virtual Table

- Provides security by **data hiding** from specific user.

Views: A Virtual Table

- Provides security by **data hiding** from specific user.
- Code can be **re-used**.

Views: A Virtual Table

- Provides security by **data hiding** from specific user.
- Code can be **re-used**.
- Views, in general, do not impact storage (But materialized views do have)

Views: DML through view

Following conditions must be met:

1. The from clause has **only one database relation**.

Views: DML through view

Following conditions must be met:

1. The from clause has **only one database relation**.
2. The select clause contains only attribute names of the relation and **does not** have any **expressions, aggregates, or distinct** specification. (Example?)

Views: DML through view

Following conditions must be met:

1. The from clause has **only one database relation**.
2. The select clause contains only attribute names of the relation and **does not** have any **expressions, aggregates, or distinct** specification. (Example?)
3. Any attribute not listed in the select clause can be set to null; that is, it does not have a **not null** constraint and is not part of a primary key.

Views: DML through view

Following conditions must be met:

1. The from clause has **only one database relation**.
2. The select clause contains only attribute names of the relation and **does not** have any **expressions, aggregates, or distinct** specification. (Example?)
3. Any attribute not listed in the select clause can be set to null; that is, it does not have a **not null** constraint and is not part of a primary key.
4. The query does not have a group by or having clause.(i.e. aggregation, falls in condition 2)

Mapping Cardinality

Ideally there are 4 types of mapping cardinality:

1. One to One

Mapping Cardinality

Ideally there are 4 types of mapping cardinality:

1. One to One
2. One to Many

Mapping Cardinality

Ideally there are 4 types of mapping cardinality:

1. One to One
2. One to Many
3. Many to One

Mapping Cardinality

Ideally there are 4 types of mapping cardinality:

1. One to One
2. One to Many
3. Many to One
4. Many to Many

Mapping Cardinality

Ideally there are 4 types of mapping cardinality:

1. One to One
2. One to Many
3. Many to One
4. Many to Many

Mapping Cardinality

Ideally there are 4 types of mapping cardinality:

1. One to One
2. One to Many
3. Many to One
4. Many to Many

2 and 3 are similar with the change of entity orientation.

Mapping Cardinality (Cont.)

- One to Many: **Foreign key** in the Many Entity

Mapping Cardinality (Cont.)

- One to Many: **Foreign key** in the Many Entity
- One to One: **Foreign key** in the Many Entity and **Unique** (similar) **Constraint** on the same key

Mapping Cardinality (Cont.)

- One to Many: **Foreign key** in the Many Entity
- One to One: **Foreign key** in the Many Entity and **Unique** (similar) **Constraint** on the same key
- Many to Many: A **junction table** is formed by combining 2 foreign keys and additional attributes (if needed)

Joins

- Natural (Inner) Join

Joins

- Natural (Inner) Join
- Outer Join (left, right, full)

SQL Code Demo

```
CREATE OR REPLACE FUNCTION totalteachers()
RETURN number IS
total number(2) := 0;
BEGIN
SELECT count(*) into total
FROM teachers;
RETURN nvl(total,-1);
END;
```