DATABASE SYSTEMS

Dr. Noha Nagy

Lecture 3

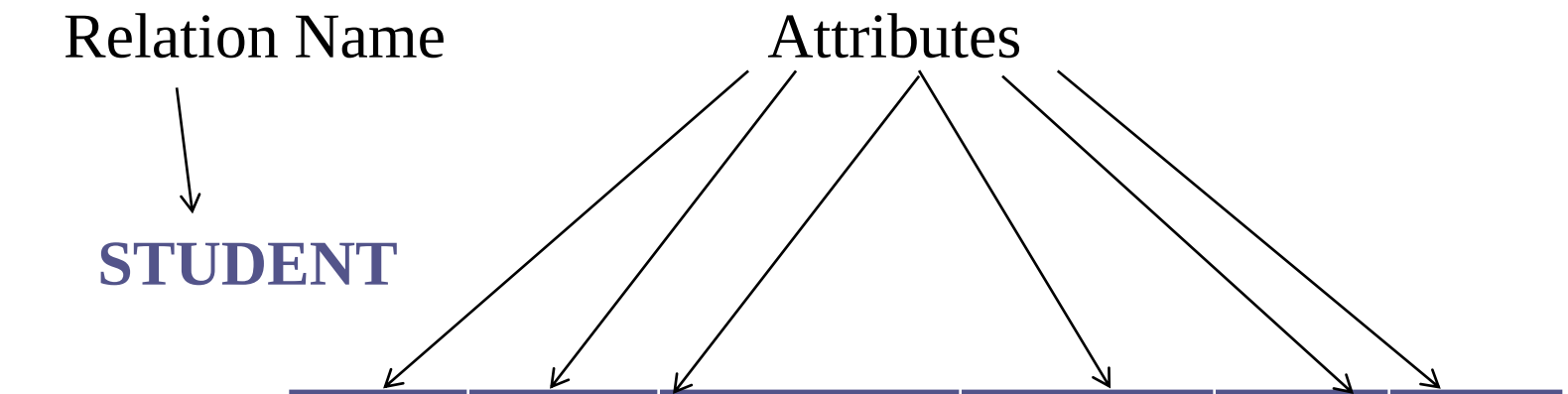
Database Constraints

Data Model

- A collection of concepts that can be used to define the DB structure
  - Data items & types
  - Relationships
  - Constraints

Relational Model

Relational Model



ST-ID	Name	National-ID	Address	Major	GPA
199	Ali	9876543	Dokki	DS	2.7
156	Maha	8765432	Nasr City	CS	3.7
789	Ali	7654321	Dokki	DS	2.7
111	Ahmad	1234567	Alex	NULL	NULL

Schema

Student(ST-ID, Name, National-ID, Address, Major,GPA)

Attributes have ATOMIC values

TRANSCRIPT 2

ST-ID	COURSE	GRADE
123	Databae-1	B
456	Compliers	C
123	Operating Systems	A
789	Compilers	B

TRANSCRIPT 1

ST-ID	COURSE	GRADE
123	Databae-1	B
	Operating Systems	A
456	Compliers	C
789	Compilers	B

TRANSCRIPT2 is not correct as the attribute COURSE is NOT ATOMIC

- **It is set valued.**

Attributes have ATOMIC values

TRANSCRIPT3

ST-ID	COURSE	GRADE
123	Databae-1, Operating Systems	B, A
456	Compliers	C
789	Compilers	B

- **Attribute COURSE is ATOMIC VARCHAR(100)**
- **Answering Queries:**
  - **Find the grade of student 123 in course Operating Systems.**
  - **Find Course where the grade of student 123 is A in these courses**

Characteristics of Relations

- Tuples have no particular order
- Ordering of attributes not important
- All values belonging to a particular attribute are from the same domain
- Attributes are atomic. (Cannot put multiple values in the same cell)
- Attributes may have a null value
  - Null is not a zero and is not an empty string

Types of Constraints

- Domain Constraints
- Key Constraints
- Integrity Constraints
  - Entity Integrity Constraint
  - Referential Integrity Constraint
  - Semantic Integrity Constraint

Domain Constraints

- The value of each attribute, A, must be an atomic value from the domain of A.

- Example, list of governorates in Egypt.
- An attribute named “Governorate” should have a value that maps to one of the governorates in that list.

Key Constraints

- Value of a key uniquely identifies a tuple in a relation
- Super key (K): is a subset of attributes of R that can identify a tuple. It might contain more than one attribute.
  - no 2 tuples have same values for K

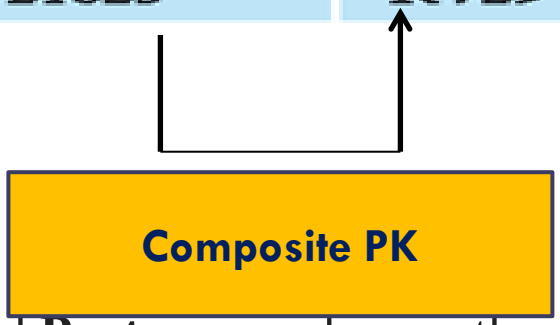
Key Constraints (Cont'd)

- A **key** is a minimal superkey; a superkey from which we cannot remove any attributes and still be able to uniquely identify tuples in a relation.
- A relational schema may have more than one key:
  - Each **key** is called a ***candidate key***
  - One designated as the ***primary key***

Examples from Premier Database – Primary Key

OrderLine

OrderNum	PartNum	NumOrdered	QuotedPrice
21608	AT94	11	$21.95
21610	DR93	1	$495.00
21610	DW11	1	$399.99
21613	KL62	4	$329.95
21614	KT03	2	$595.00
21617	BV06	2	$794.95
21617	CD52	4	$150.00
21619	DR93	1	$495.00
21623	KV29	2	$1,290.00



Ordernum and **Partnum** makes up the primary key Of the OrderLine table. This is what is known as a **Composite Primary key**, that is, primary key that is made up of more than one field.

Integrity Constraints

- Integrity constraints are specified on a schema and hold for every instance of the schema

- Entity integrity constraint
  - A primary key value cannot be null
- Referential integrity constraint
  - If R_1 refers to R_2 then $t_1 \in r_1(R_1)$ must refer to an existing $t_2 \in r_2(R_2)$

Foreign Key

- **A Foreign key is how we relate relations to each other**
- **A Foreign key in R is a set of attributes FK in R such that FK is a primary key of some other relation R**
- **A foreign key is used to specify a referential integrity constraint.**

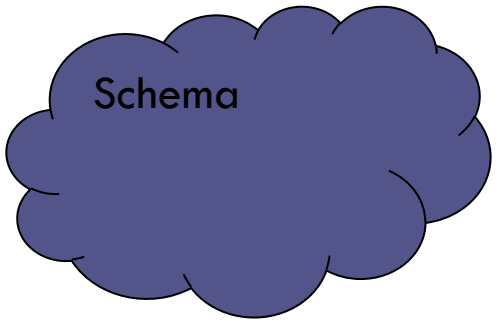
Example

Employee

Enum	Ename	phone	projectnum

Project

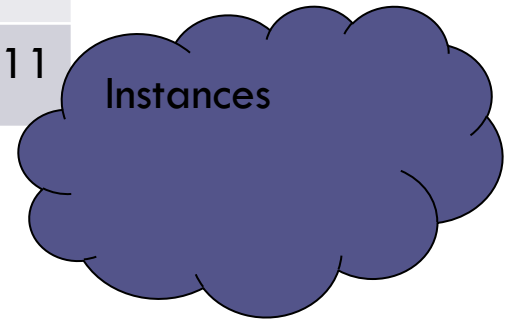
Pnum	Pname	Location



Example

Employee

Enum	Ename	phone	Projectnum
123	Ahmed	01110025878	111
124	Ali	01225929785	
127	Ola	0102457896	111



Project

Pnum	Pname	Location
111	ABC	Giza
112	EFG	Cairo

3-Integrity Constraints

- Integrity constraints are specified on a schema and hold for every instance of the schema
- **Entity integrity constraint**
  - **Primary key**
    - Unique
    - Not null
    - Stable and minimum # of attributes
- **Referential integrity constraint**
  - **Foreign key**
    - Match a primary key
    - Null

Semantic Integrity Constraints

- Constraints on data values
- Examples:
  - The salary of an employee **must not exceed** that of her supervisor.
  - The total of available seats **must be > 0** in order for a reservation to be made.
  - A person's date of birth **must be before** the current date.

Operations on Relations

- Operations include insert, delete, modify (update), and retrieval.
- Some operations can violate database constraints.

Insert Operation

- Provide a list of attribute values to be inserted (ie. A new tuple)
- Example
  - Insert into Employee values (Mary, 0102457896)

Employee

Enum	Ename	phone
123	Ahmed	01110025878
124	Ali	01225929785
	Mary	**0102457896**

Insert Operation

- Inserts may violate constraints
  - Entity Integrity Constraint: : Inserting a tuple without the primary key
  - Referential Integrity Constraint: Inserting a tuple with a foreign key that is not available in the main relation. Example adding an employee in a department number that does not exist.

Example

Employee

Enum	Ename	phone	Projectnum
123	Ahmed	01110025878	111
124	Ali	01225929785	
127	Ola	0102457896	115

Project

Pnum	Pname	Location
111	ABC	Giza
112	EFG	Cairo

Delete Operation

- Deletes a tuple from a relation
- Example:
  - Delete a tuple with ID=123456
- The only constraint which can be violated is the referential integrity constraint (ie. A tuple in another relation references the tuple that is stated for deletion). If that employee works on some projects and is available as a foreign key the other relation named “Project”, the deletion of that employee will violate the referential integrity constraint.

Modify Operation

- Change the value for one or more attributes in a relation
- Example:
  - Modify salary of employee 123 to 20000
- Modifying a primary key is like deleting a tuple and adding a new one (same violations may apply)

SQL
[STRUCTURED QUERY LANGUAGE]

SQL

- *Structured Query language* **SQL** is pronounced as “S-Q-L” or sometimes as “See-Quel” which is the standard language for dealing with **Relational Databases.**
- It is effectively used to create, insert, search, update, delete, modify database records.

A Few Details

- SQL **commands** are case insensitive:
  - Same: SELECT, Select, select
  - Same: Product, product

- Use single quotes for constants:
  - 'abc' - yes
  - "abc" - no

SQL is a…

- **Data Definition Language (DDL)**
  - **Define relational *schemata***
  - **Create/alter/delete tables and their attributes**
- **Data Manipulation Language (DML)**
  - **Insert/delete/modify tuples in tables**
  - **Query one or more table**
- **Data Control Language (DCL)**
  - **Specify user permissions**
  - **Grant/revoke**