Database Systems (CS F212)- Lecture 5 - 6

BITS Pilani
Pilani Campus

Recapitulation of Lecture 4

- Database languages
- Database system architecture
  - Database user
  - Query processor
  - Storage Manager/ Database Manager
  - Data Storage

Relational Database

Relational Database:

- A database that stores data in a structured format, using rows and columns

Why relational?

- Data stored are related to each other

RDBMS

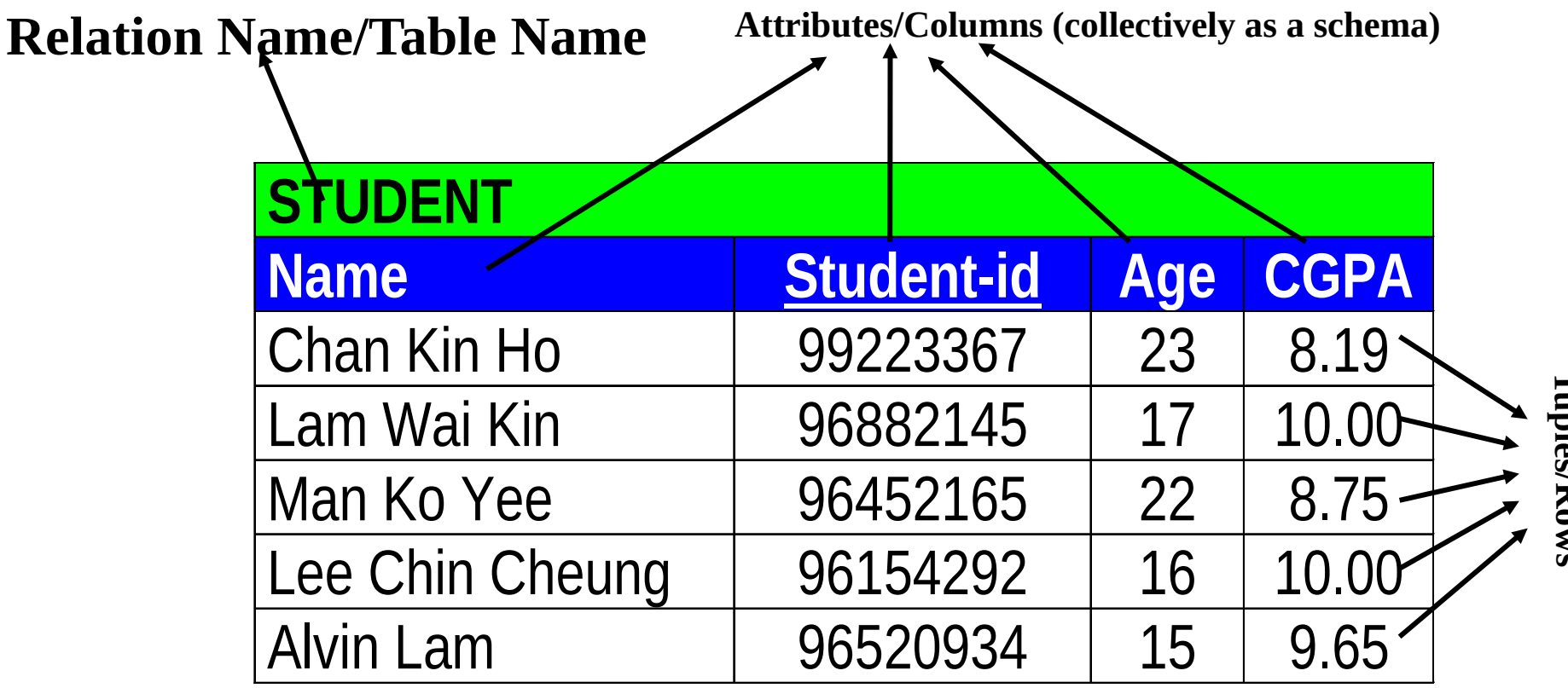
- Relational Database Management System
- Program to manage relational databases

Relational Database

Terminologies used in RDB:

- **Relations:** Tables in the DB
- **Tuples:** Rows in a relation
- **Attributes:** Columns in a relation
- **Cardinality:** No. of rows in a relation
- **Degree/arity:** No. of columns in a relation
- **Domain of an attribute:** Set of permitted values
- **Atomic values:** Values are indivisible units

Relational Database



STUDENT			
Name	Student-id	Age	CGPA
Chan Kin Ho	99223367	23	8.19
Lam Wai Kin	96882145	17	10.00
Man Ko Yee	96452165	22	8.75
Lee Chin Cheung	96154292	16	10.00
Alvin Lam	96520934	15	9.65

- Cardinality = 5, degree = 4, all rows distinct

Relations defined Mathematically

Relation:

- Defined on a collection of domains D_1, D_2,..., D_n
- Consists of two parts: a "heading" and a "body"

Heading:

- A fixed set of attribute-domain pairs { $(A_1:D_1)$, $(A_2:D_2)$, ... , (A_n,D_n) }
- Also known as **schema**

Body:

- A time-varying set of tuples
- Each tuple consists of a set of attribute-value pairs

{ $(A_1:v_i^1)$, $(A_2:v_i^2)$,..., $(A_n:v_i^n)$ } and i = 1, 2, ..., m

- Also known as **instance**

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Instructor

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

Department

- Using common attributes in relation schemas is one way or relating tuples of distinct relations

- *Find all instructor names who work in Watson building*

course_id	sec_id	semester	year	building	room_number	time_slot_id
BIO-101	1	Summer	2009	Painter	514	B
BIO-301	1	Summer	2010	Painter	514	A
CS-101	1	Fall	2009	Packard	101	H
CS-101	1	Spring	2010	Packard	101	F
CS-190	1	Spring	2009	Taylor	3128	E
CS-190	2	Spring	2009	Taylor	3128	A
CS-315	1	Spring	2010	Watson	120	D
CS-319	1	Spring	2010	Watson	100	B
CS-319	2	Spring	2010	Taylor	3128	C
CS-347	1	Fall	2009	Taylor	3128	A
EE-181	1	Spring	2009	Taylor	3128	C
FIN-201	1	Spring	2010	Packard	101	B
HIS-351	1	Spring	2010	Painter	514	C
MU-199	1	Spring	2010	Packard	101	D
PHY-101	1	Fall	2009	Watson	100	A

Course_Detail

Find the relation name, attributes, tuples, cardinality , degree, relation schema

Key Constraints and Constraints on Null Value

Superkey: Set of one or more attributes that, taken collectively, uniquely identifies a tuple in the relation

If R denotes the set of *attributes in a relation r and subset K of R is identified to be a superkey for r, then for two distinct tuples t1 and t2 cannot have the same values on* all attributes in *K.*

Example: *student (ID, name, dept name, CGPA)*
Valid superkey are {ID, name}, {ID, name}*, etc.*

Any set K' , such that, K ⊆ K *'* ⊆ R, *is a valid superkey*

Keys in a Database

Keys: An attribute or set of attributes which satisfies the following properties :

- Two distinct tuples in any state of the relation cannot have identical values for (all) the attributes in the keywhich helps in identifying a *tuple* in a relation
- It is a minimal super key: that is a superkey from which we cannot remove any attributes and still have the uniqueness constraint condition in 1 hold

Candidate Key

- Key is determined based on the meaning of the attributes, and the property is *time-invariant*
- **Candidate Key:**
  - Set of keys which can uniquely identify a record in a relation
  - Minimal superkey

Example:

instructor*(ID, name, dept name, salary)*

Superkey *= {{ID, name}, {ID, name, dept}, .. }*

candidate key *= {ID, {name+dept_name}}*

Primary Key

Primary Key: Candidate key chosen by the DB designers to uniquely identify a record in a relation

- Attributes that forms the primary key are underlined

Key is the property of entire relation.

- It is suggested to use a Key having a single attribute as primary key

- Represented by underlining the column name

- **Alternate key :** All the keys which are not primary key are called an alternate key

Primary Key

Example:

instructor*(ID, name, dept name, salary)*

candidate key *= {ID, {name+dept_name}}*
Primary key = *ID*
Alternate Key *= dept+name*

Foreign Key

- Used to establish relationship between two relations

- ***A set of attributes FK in relation schema R_1 is a foreign key of R_1 that reference relation R_2 if it satisfies the following rules:***
  1. The attributes in FK have the same domaint(s) as the primary key attributes PK of R_2; the attributes FK are said to **reference** or **refer to** the relation R_2.

2. A value of FK in a tuple t_1 of the current state $r(R_1)$ either occurs as a value of PK for some tuple t_2 in the current state $r_2(R_2)$ or is null. In the former case, we have $t_1[FK] = t_2[PK]$, and we say that the tuple t_1

Foreign Key

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

Instructor

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

Department

- FK is *dept_name* in Instructor relation and it refers to *dept_name* column in Department table
- Department and Instructor relations are known as **Referenced/Parent** and **Referencing/Child** relations, respectively.

Foreign Key

- **Self-referential foreign key:** Foreign key references a parent key in the same table.

employee(ID, name, dept_name, reports_to)

If *reports_to* contains the ID of the person to whom an employee reports to, then for the above given relation

Primary key = {ID}
Foreign key = {reports_to}

Integrity Constraints in Relational Database

Types of integrity constraints in relational database

1. **Domain integrity constraint**
2. **Entity integrity constraint**
3. **Referential integrity constraint**

- **Domain integrity constraint:**

•All columns in a relational database must be declared upon a defined domain

Student_detail(ID varchar(10), Name varchar(10))

Integrity Constraints in Relational Database

- **Entity integrity constraint:** Primary key value can't be *Null*
- **Referential integrity constraint:**
  - Specified between two relations
  - Used to maintain consistency among the tuples in the two relation
  - A tuple in one relation that refers to another relation must refer to an existing tuple in that relation

Integrity Constraints in Relational Database

Customer

Customer_ID	Customer_name
1	Tom
2	John
Null	Tom

Entity integrity constraint violated

Customer

Customer_ID	Customer_name
1	Tom
2	John
3	Tom

Order

Order_ID	C_ID	Order_date
1	1	12-01-2020
2	4	12-01-2020
3	2	13-01-2020

Referential integrity constraint violated