15CSE302 Database Management Systems

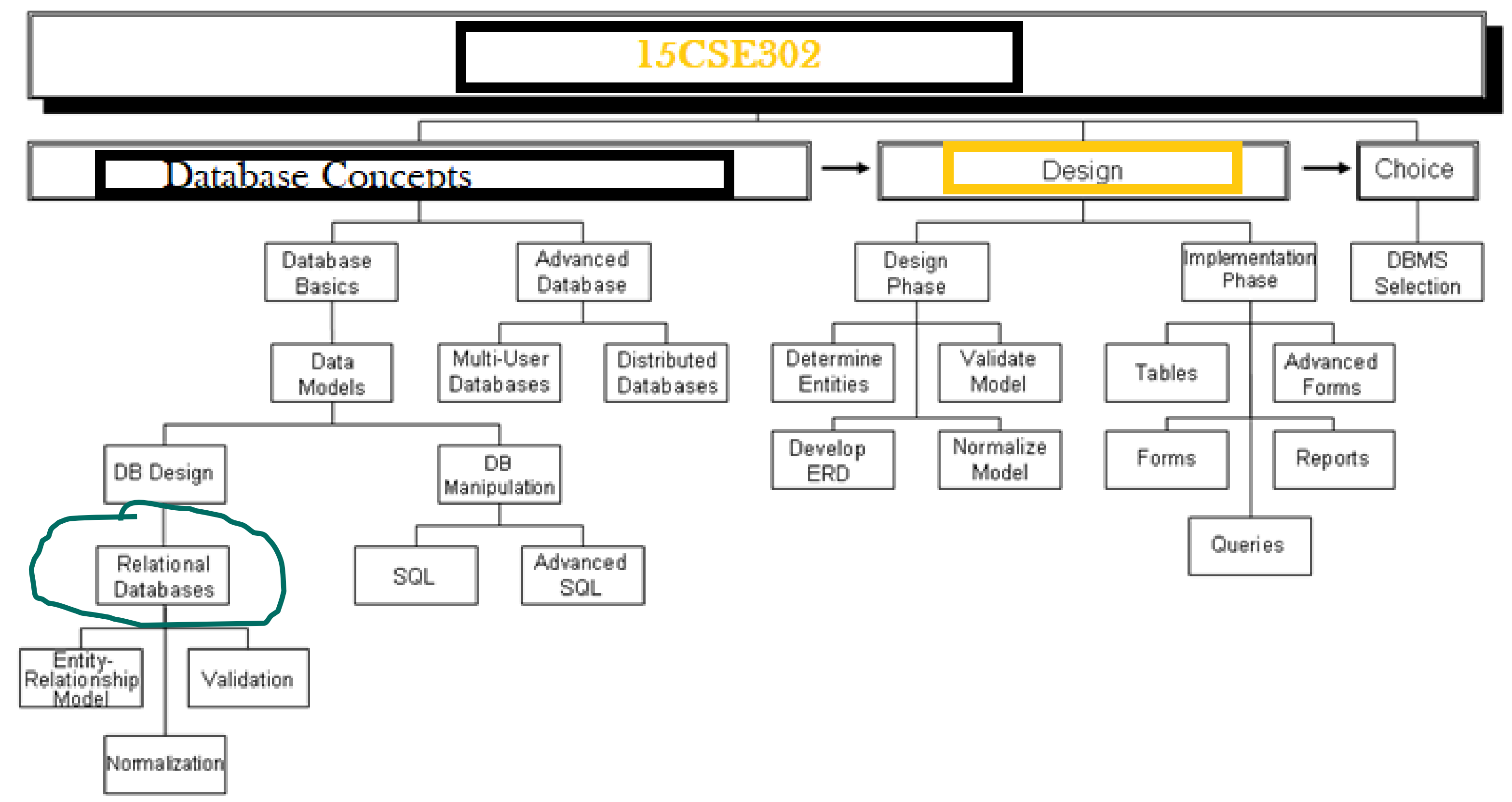
Lecture 3 Relational Model

B.Tech /III Year CSE/V Semester

L T P C 2 0 2 3

DBMS Team
Dr G Jeyakumar
Bindu K R
Dr Priyanka Kumar
R. Manjusha
Department of CSE
Amrita School of Engineering

Syllabus


15CSE302
Database Concepts
Design
Choice
Database Basics
Advanced Database
Design Phase
Implementation Phase
DBMS Selection
Data Models
Multi-User Databases
Distributed Databases
Determine Entities
Validate Model
Tables
Advanced Forms
DB Design
DB Manipulation
Develop ERD
Normalize Model
Forms
Reports
Relational Databases
SQL
Advanced SQL
Queries
Entity-Relationship Model
Validation
Normalization

Brief Recap of Previous Lecture

- Database Abstraction
- Instances and Schemas
- Data Models
- Database Users

Today's Lecture

- **Structure of Relational Databases**
- **Keys**
- **Schema Diagrams**

Introduction

- Relational data model is the commercial data model for today's applications.

- It is simple and ease of use for programmer compared to other models.

Relational Database: Definitions

- Relational database: a set of relations
- Relation: made up of 2 parts:
  - Schema : specifies name of relation, plus name and type of each column.
    e.g. Students(sid: string, name: string, login: string, age: integer, gpa: real)
- Instance : a table, with rows and columns.
  - #Rows = cardinality
  - #fields = degree / arity
- Can think of a relation as a set of rows or tuples (i.e., all rows are distinct).
- Columns (attributes) are single-valued/atomic.

Relational Database: Definitions

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

- #Rows = cardinality=?
- #fields = degree / arity=?

Relational Database: Definitions

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

- #Rows = cardinality=3
- #fields = degree / arity=5
- All rows are distinct

Structure of Relational Model

attributes
(or columns)

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 Table.

tuples
(or rows)

- Relational DB is a collection of Tables or Relation.
- E.g. *Instructor* Table.
- There are four columns/attributes
- Each row is storing the values for
- ID, name, dept_name and salary of an Instructor.
- Each row is uniquely identified by ID.

- **Another Example - Refer the *Course* Table.**

course_id	*title*	*dept_name*	*credits*
BIO-101	Intro. to Biology	Biology	4
BIO-301	Genetics	Biology	4
BIO-399	Computational Biology	Biology	3
CS-101	Intro. to Computer Science	Comp. Sci.	4
CS-190	Game Design	Comp. Sci.	4
CS-315	Robotics	Comp. Sci.	3
CS-319	Image Processing	Comp. Sci.	3
CS-347	Database System Concepts	Comp. Sci.	3
EE-181	Intro. to Digital Systems	Elec. Eng.	3
FIN-201	Investment Banking	Finance	3
HIS-351	World History	History	3
MU-199	Music Video Production	Music	3
PHY-101	Physical Principles	Physics	4

- **Course Table/Relation**
- Four columns/attributes

- Each row is storing the values for
- course_id, title, dept_name and credits of a course.

- Each course is uniquely identified by its course_id.

The *prereq* relation.

course_id	*prereq_id*
BIO-301	BIO-101
BIO-399	BIO-101
CS-190	CS-101
CS-315	CS-101
CS-319	CS-101
CS-347	CS-101
EE-181	PHY-101

- **Prereq**
- **Two columns/Attributes**
- **In Each row, the second course is the prerequisite course for the first course.**
- **Thus, each row indicates two course are related.**
- **Similarly, in *Instructor* table, ID is related to name, dept_name and Salary of an instructor.**
- **and in *course* table the course_ide is related to title, dept_name and credit.**

Relational Model

- In general, table represents relationship among set of values.
- Relation/Table is a collection of relationships.
- 'Table' in Database design is similar to 'relation' in Mathematics....
- Thus the term 'Relational Model'.

- In Mathematics ...
- Tuple is a sequence of values.
- A relationship between n values is represented as and n-tuple of values.

- In Relational Model ...
- Relation refers to a table.
- Tuple refers to a row.
- Attribute refers to a column.

Relation Instance

- Refers to specific instance of a relation.
- Refer the *Instructor* relation instance, has 12 rows corresponding to 12 instructors.
- The order in which the row are arranged is irrelevant.
- It can be sorted or unsorted.
- Both the instances in right side are the same.

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

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

Concepts underlying Relational Model - *domain*

- The set of allowed values for each attribute is called the **domain** of the attribute.

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

- Domain of salary – Set of salary of instructors.
- Domain of name – Set of names of instructors.

- salary = {65000, 900000, 40000, 95000, 60000….80000}

- Each element in the domain a value for the attribute.

- 40000 is a value for the salary attribute.

Concepts underlying Relational Model - *Atomic*

- For each relation (*r*), the domain of all attributes of *r* be atomic.
- that is, indivisible.

- salary = {65000, 900000, 40000, 95000, 60000....80000}
- Is salary attribute atomic?
- Answer: yes. Why?

- Suppose we have 'Phone-Number' attribute in instructor relation, and allows multiple phone numbers for each instructor,
- A sample domain of 'Phone-Number'
- ={(0422-2685000,+91-8967563421),
- (8934765478), (1278347865, 0433-783456)......}
- First and third elements of this domain has two phone numbers, hence it is non-atomic domain.

Concepts underlying Relational Model - *Atomic*

- A sample domain of '**Phone-Number**'

{0422-2685000, 0433-783456, +91-8967452312}

- Is it Atomic domain?

Answer : No..

- Why? – each phone number is divisible – country code, city code and phone number.

- A sample domain of '**Phone-Number**'
- {3456789023, 3948576819, 7829345678}

- Is it Atomic domain?

Answer: Yes

- **Why? – each element in this domain is single value ie., indivisible.**

Concepts underlying Relational Model - NULL

- The special value ***null*** is a member of every domain. Indicated that the value is “unknown” or does not exist.
- The null value causes complications at the time of accessing the data from database, hence use of null to be restricted.

- If an instructor does not have phone, the value for his phone-number attributed to be stored ‘null’

Concepts underlying Relational Model -

Database schema Vs Database Instance.

- **Database Schema** – the logical design of the database.
- **Database Instance** – the content of the database at an instance of time.

Database Schema

STUDENT

Name	StudentNumber	Class	Major

COURSE

CourseName	CourseNumber	CreditHours	Department

SECTION

SectionIdentifier	CourseNumber	Semester	Year	Instructor

GRADE_REPORT

StudentNumber	SectionIdentifier	Grade

Database Instance

STUDENT

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	07	King
92	CS1310	Fall	07	Anderson
102	CS3320	Spring	08	Knuth
112	MATH2410	Fall	08	Chang
119	CS1310	Fall	08	Anderson
135	CS3380	Fall	08	Stone

GRADE_REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

Concepts underlying Relational Model -
relation Schema Vs relation instance

Relation Schemas

STUDENT

Name	StudentNumber	Class	Major

COURSE

CourseName	CourseNumber	CreditHours	Department

SECTION

SectionIdentifier	CourseNumber	Semester	Year	Instructor

Relation Instance

STUDENT

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	07	King
92	CS1310	Fall	07	Anderson
102	CS3320	Spring	08	Knuth
112	MATH2410	Fall	08	Chang
119	CS1310	Fall	08	Anderson
135	CS3380	Fall	08	Stone

Concepts underlying Relational Model -
relation schema & instance...

- int marks [5]; ----- Variable
- marks[ ]=[78, 93, 90, 89, 83] ----- values

- Instance – changes.
- Schema – does not.

Let us proceed further with University Database

University Database – Relation Schemas

classroom(building, room_number, capacity)
department(dept_name, building, budget)
course(course_id, title, dept_name, credits)
instructor(ID, name, dept_name, salary)
section(course_id, sec_id, semester, year, building, room_number, time_slot_id)
teaches(ID, course_id, sec_id, semester, year)
student(ID, name, dept_name, tot_cred)
takes(ID, course_id, sec_id, semester, year, grade)
advisor(s_ID, i_ID)
time_slot(time_slot_id, day, start_time, end_time)
prereq(course_id, prereq_id)

University Database – Relation Instances

instructor

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

course

course_id	title	dept_name	credits
BIO-101	Intro. to Biology	Biology	4
BIO-301	Genetics	Biology	4
BIO-399	Computational Biology	Biology	3
CS-101	Intro. to Computer Science	Comp. Sci.	4
CS-190	Game Design	Comp. Sci.	4
CS-315	Robotics	Comp. Sci.	3
CS-319	Image Processing	Comp. Sci.	3
CS-347	Database System Concepts	Comp. Sci.	3
EE-181	Intro. to Digital Systems	Elec. Eng.	3
FIN-201	Investment Banking	Finance	3
HIS-351	World History	History	3
MU-199	Music Video Production	Music	3
PHY-101	Physical Principles	Physics	4

prerequisite

course_id	prereq_id
BIO-301	BIO-101
BIO-399	BIO-101
CS-190	CS-101
CS-315	CS-101
CS-319	CS-101
CS-347	CS-101
EE-181	PHY-101

department

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

section

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

teaches

ID	course_id	sec_id	semester	year
10101	CS-101	1	Fall	2009
10101	CS-315	1	Spring	2010
10101	CS-347	1	Fall	2009
12121	FIN-201	1	Spring	2010
15151	MU-199	1	Spring	2010
22222	PHY-101	1	Fall	2009
32343	HIS-351	1	Spring	2010
45565	CS-101	1	Spring	2010
45565	CS-319	1	Spring	2010
76766	BIO-101	1	Summer	2009
76766	BIO-301	1	Summer	2010
83821	CS-190	1	Spring	2009
83821	CS-190	2	Spring	2009
83821	CS-319	2	Spring	2010
98345	EE-181	1	Spring	2009

Instructor is related to *department* by dept_name attribute.

Instructor is related to *sections* by course_id, sec_id and semester and the association is stored in a new relation *teaches*.

. . .

Relation Schema and Instance

- If $A_1, A_2, \ldots, A_n$ are attributes

- $R = (A_1, A_2, \ldots, A_n)$ is a relation schema
  Example:
  instructor = (ID, name, dept_name, salary)

- Formally, given sets $D_1, D_2, \ldots. D_n$ which are the domains $A_1, A_2, \ldots, A_n$, a relation r is a subset of $D_1 \times D_2 \times \ldots \times D_n$.

- Thus, a relation is a set of n-tuples $(a_1, a_2, \ldots, a_n)$ where each $a_i \in D_i$

- Structure of Relational Databases
- Keys
- Schema Diagrams

Keys

- **No two relations in a relation is allowed to have same values in all the attributes.**

Roll Number	Name
CB.EN.U4CSE18201	AADURU VENKATA HEMA ABHINAV.
CB.EN.U4CSE18202	S.AAKASH MUTHIAH.
CB.EN.U4CSE18203	ABISHEK VASANTHAN A S.
CB.EN.U4CSE18204	M. S. ADARSH.
CB.EN.U4CSE18205	ADITHI NARAYAN.
CB.EN.U4CSE18206	AMBATI NAGA SREEHARSHA REDDY.

- **Tuples with in a given relation to be distinguished, by the values of the attributes.**

- ❑ Super Key
- ❑ Candidate Key
- ❑ Primary Key
- ❑ Foreign Key

Superkey

- Let $K \subseteq R$
- *K* is a **superkey** of *R* if values for *K* are sufficient to identify a unique tuple of each possible relation *r(R)*

- Example: {*ID*} and {ID, name} are both superkeys of *instructor.*

- *{name}* is not a superkey, because more instructors may have same name.

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

Let R denote the set of attributes in the schema of relation r.
For a subset of K of R is a superkey for r no distinct tuples have the same values for all attributes in K.
i.e. If t1 and t2 are in r and t1<> t2, t1.K<>t2.K

Candidate Key

- A superkey may have extraneous attribtute.
- Eg. In {ID, name}, name is extraneous.

- Superkey for which no proper subset is a superkey is the candidate key.
- Superkey *K* is a **candidate key** if *K* is minimal
  Example: {*ID*} is a candidate key for *Instructor, but not* {ID, name}.

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

Many candidate keys for a relation
Eg. {ID} and {name, dept_name} for *instructor*.

Entity Integrity- Primary key

- Several distinct set of attributes could serve as candidate keys.

- The candidate key which is primarily chosen by the database designer is the **primary key.**
  {ID} in instructor is the primary 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

Entity Integrity -Primary Key

- Must be chose with care.

- Their attribute values are never or very rarely changed.
  Eg. RollNumber.

- The primary key of a relation to be listed before other attributes.

  Refer the *department* relation.

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

Referential Integrity -Foreign Keys

- A relation, say r_1, may include among its attributed the primary key of another relation, say r_2.
- This attribute is called as foreign key from r_1, referencing to r_2.
- The relation r_1 is called as the referencing relation and r_2 is called the referenced relation.
- The foreign key forms a dependency between r_1 and r_2.
- Eg. *dept_name* attribute, from *instructor* relation to *department* relation.

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

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

In any database instance, given any tuple, say t_a, from the *instructor* relation, there must be some tuple, say t_b, in the *department* relation such that the value of the *dept_name* attribute of t_a is the same as the value of the primary key, *dept_name*, of t_b.

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

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

Referential Integrity Constraints- Foreign Key

- If a section exists for a course, it must be taught by at least one instructor.
- However, it could possibly be taught by more than one instructor.....This is a constraint.

- To enforce above constraint,
- we require that if a particular (*course_id, sec_id, semester, year*) combination appears in *section*
- then the same combination must appears in *teaches*. (Looks Like Foreign key from section to teaches).
- However this set of value is not form a primary key for *teaches*, since more than one instructor may teach one such section.
- So we can not set foreign key from section to teaches
- **(however we can define from teaches to section … why?)**

section

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

ID	*course_id*	*sec_id*	*semester*	*year*
10101	CS-101	1	Fall	2009
10101	CS-315	1	Spring	2010
10101	CS-347	1	Fall	2009
12121	FIN-201	1	Spring	2010
15151	MU-199	1	Spring	2010
22222	PHY-101	1	Fall	2009
32343	HIS-351	1	Spring	2010
45565	CS-101	1	Spring	2010
45565	CS-319	1	Spring	2010
76766	BIO-101	1	Summer	2009
76766	BIO-301	1	Summer	2010
83821	CS-190	1	Spring	2009
83821	CS-190	2	Spring	2009
83821	CS-319	2	Spring	2010
98345	EE-181	1	Spring	2009

teaches

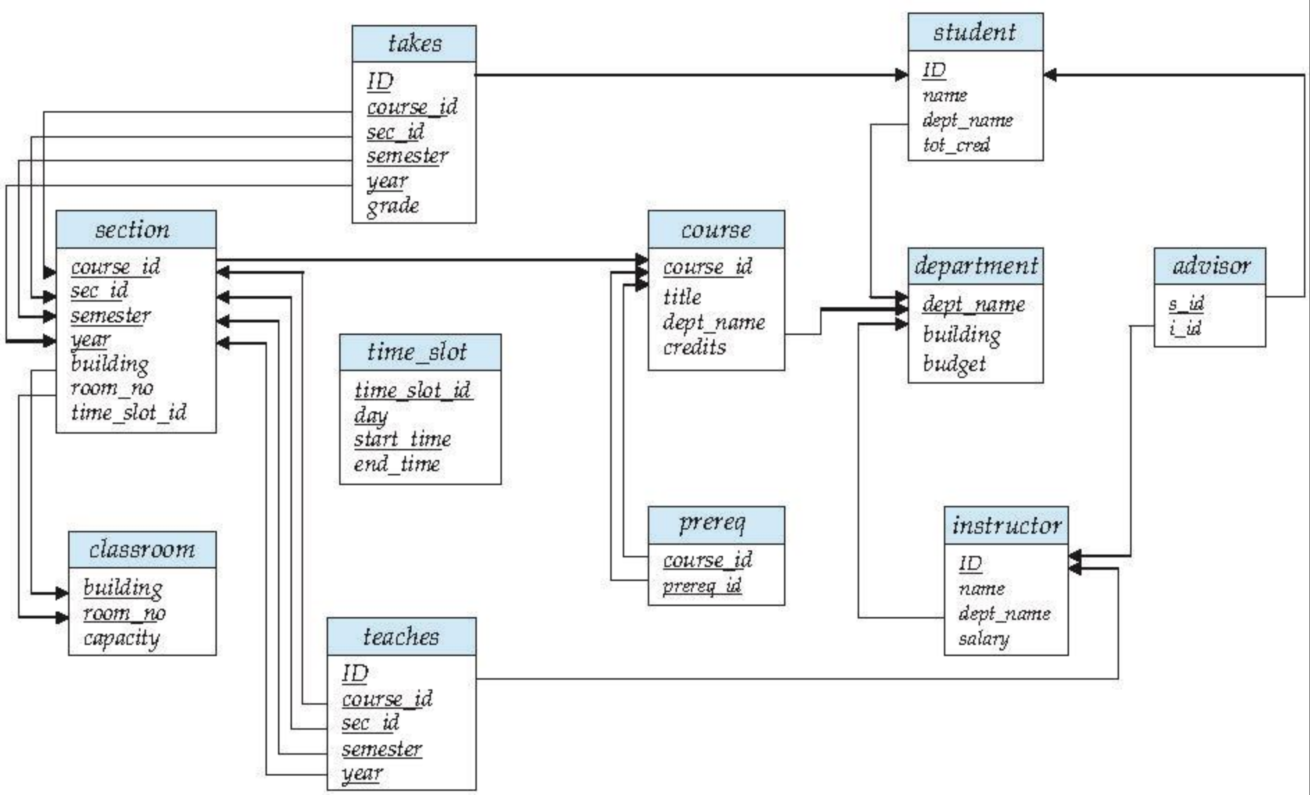
Referential Integrity Constraints

- The constraint from *section* to *teaches* is an example of a referential integrity constraint.
- A referential integrity constraint requires that the value appearing in the specified attributes of any tuple in the referencing relation also appears in specified attributes of at least one tuple in the referenced relation.
- Example: (Refer relations in the previous slide)
- section Referencing teaches by (course_id, sec_id, semester, year)
- **section is referencing relation**
- **teaches is referenced relation**

- Structure of Relational Databases
- Keys
- Schema Diagrams

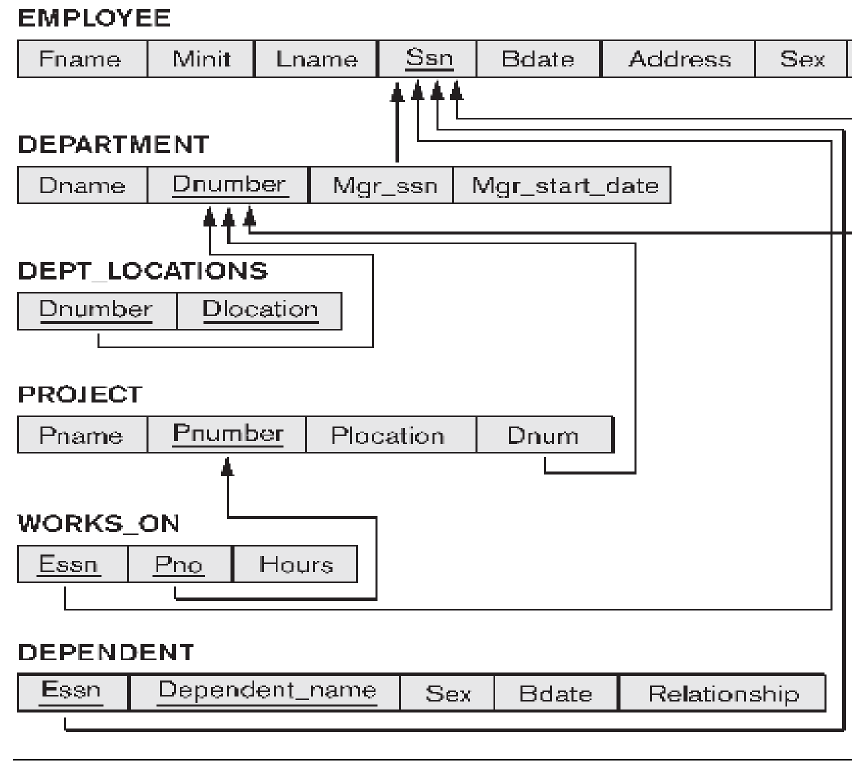
Schema Diagram

- **The database schema and its keys can be visualized in Schema Diagram.**



Schema Diagram

- **The database schema and its keys can be visualized in Schema Diagram.**



Company database

Summary

- Structure of Relational Databases
- Keys
- Schema Diagrams

Next Lecture

- Relational Query Languages
- Relational Algebra

References

- https://docs.oracle.com/en/database/oracle/oracle-database/20/newft/new-features.html
- https://www.db-book.com/db7/index.html

Thank You

Happy to answer any questions ! ! !