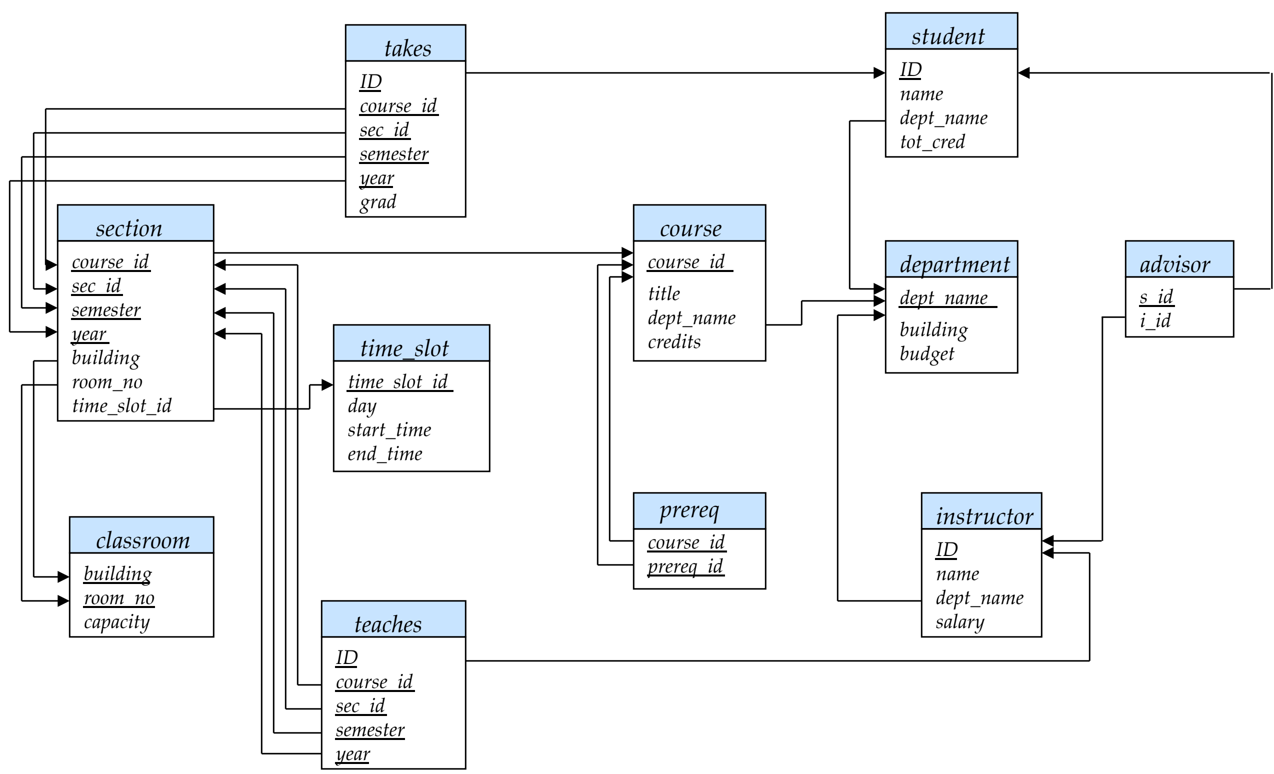
1 Database Schema Diagrams

We will use the university sample relational schema defined in lectures. The following schema diagram illustrates all the relations in the database. Each relation has primary keys underlined. Draw arrows to show all *foreign key* constraints.



2 Relational Algebra

We use relational algebra to specify *queries* on a database. This is the formal mathematical notational. Later we will see how this translates into SQL.

Summary of Relational Algebra

[1] Diagrams and examples taken from Database System Concepts 6th Edition

Symbol (Name)	Example of Use
σ (Selection)	$\sigma_{\text{salary}>=85000}(instructor)$
	Return rows of the input relation that satisfy the predicate.
Π (Projection)	$\Pi_{ID,salary}(instructor)$
	Output specified attributes from all rows of the input relation. Remove duplicate tuples from the output.
$\bowtie$ (Natural join)	$instructor \bowtie department$
	Output pairs of rows from the two input relations that have the same value on all attributes that have the same name.
$\times$ (Cartesian product)	$instructor \times department$
	Output all pairs of rows from the two input relations (regardless of whether or not they have the same values on common attributes)
$\cup$ (Union)	$\Pi_{name}(instructor) \cup \Pi_{name}(student)$
	Output the union of tuples from the two input relations.

2.1 Lets practice...

The university schema and relations are at the end of this handout.

1. What is the result of $\sigma_{s\_id=ID}(\text{student} \times \text{advisor})$?
2. Explain what each of the following expressions do:
   (a) $\sigma_{year \geq 2009}(\text{takes}) \bowtie \text{student}$
   (b) $\sigma_{year \geq 2009}(\text{takes} \bowtie \text{student})$
   (c) $\Pi_{\text{ID, name, course\_id}}(\text{student} \bowtie \text{takes})$
3. Suppose we have the following relational database:

   employee (person_name, street, city)
   works(person_name, company_name, salary)
   company (company_name, city)

   (a) Find the names of all employees who live in city 'Miami'.

   $$\Pi_{person\_name}(\sigma_{city=\text{'Miami'}}(employee))$$

   (b) Find the names of all employees whose salary is greater than \$100,000.

   $$\Pi_{person\_name}(\sigma_{salary>100000}(works))$$

(c) Find the names of all employees who live in 'Miami' and whose salary is greater than \$100,000.

$$\Pi_{person\_name}(\sigma_{city=\text{'Miami'}\wedge salary>100000}(employee \bowtie works))$$

(d) Find the names of all employees who work for "First Bank Corporation".

$$\Pi_{person\_name}(\sigma_{company\_name=\text{'First Bank Corporation'}}(works))$$

(e) Find the names and cities of residence of all employees who work for "First Bank Corporation".

$$\Pi_{person\_name,city}(\sigma_{company\_name=\text{'First Bank Corporation'}}(works \bowtie employee))$$

(f) Find the names, street address, and cities of residence of all employees who work for "First Bank Corporation" and earn more than \$10,000.

$$\Pi_{person\_name,street,city}(\sigma_{company\_name=\text{'First Bank Corporation'}\wedge salary>10000}(employee \bowtie works))$$

4. Suppose we have the following relational database:

branch(branch_name, branch_city, assets)
customer(customer_name, customer_street, custromer_city)
loan(loan_number, branch_name, amount)
borrower(customer_name, loan_number)
account(account_number, branch_name, balance)
depositor(custromer_name, account_number)

(a) Underline appropriate primary keys.

Primary keys are bold font. Note that some of this is up to interpretation of the database - feel free to discuss and come to some agreement. A sample solution might be:

*branch(**branch_name**, branch_city, assets)*
*customer(**customer_name**, **customer_street**, customer_city) ** customer_names may not be unique***
*loan(**loan_number**, branch_name, amount) ** assume that loan numbers are across all branches*
*borrower(**customer_name**, **loan_number**)*
*account(**account_number**, branch_name, balance)*
*depositor(**customer_name**, **account_number**)*

(b) Given your choice of primary keys, identify appropriate foreign keys. Draw a relation diagram to indicate foreign keys.

try having the students draw a box for each schema and then draw the relation diagram for the foreign keys as is done on the diagram on the first page.

*branch(**branch_name**, branch_city, assets)*
*customer(**customer_name**, **customer_street**, customer_city) ** names may not be unique***
*loan(**loan_number**, branch_name, amount) ** assume that loan numbers are across all branches*
*borrower(**customer_name**, **loan_number**)*
*account(**account_number**, branch_name, balance)*
*depositor(**customer_name**, **account_number**)*

Foreign Keys:

loan and *account* have foreign key *branch_name* for *branch.*

borrower has foreign key *loan_number* for *loan.*
depositor has foreign key *account_number* for *account.*

(c) Find the names of all branches located in 'Chicago'.
$\Pi_{branch\_name}(\sigma_{branch\_city=\text{'Chicago'}}(branch))$

(d) Find the names of all borrowers who have a loan in branch 'Down-town'.

$\Pi_{customer\_name}(\sigma_{branch\_name=\text{'Down-town'}}(borrower \bowtie loan))$

(e) Find all loan numbers with a loan value greater than \$10, 000.

$\Pi_{loan\_number}(\sigma_{amount>10000}(loan))$

(f) Find the names of all depositors who have an account with a value greater than \$6000.

$\Pi_{customer\_name}(\sigma_{balance>6000}(depositor \bowtie account))$

3 University Relations

Relations and their 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)

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

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

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

Section

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

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

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