RELATIONS AND RELATIONAL ALGEBRA -v2

Introduction to Database Systems
Mahdi Akhi
Sharif University of Technology

IN THIS LECTURE

- The Relational Model
  - Relational data structure
  - Relational data manipulation
- For more information
  - Connolly and Begg – Chapters 3.1-3.2.2 and 4
  - Ullman and Widom – Chapter 3.1, 5.1
  - Previous Version of RELATIONS AND RELATIONAL ALGEBRA Slide

Relation - Structure

Mathematical Definition

Consider m distinct attributes $A_1, A_2, \ldots, A_m$ each having a domain $D_1, D_2, \ldots, D_m$ (not necessarily distinct) respectively.

Simply, R is defined as a set of two underlying sets:

1. **Heading.** $\{A_1, \ldots, A_m\}$, denoted by H_R .
2. **Body.** Any subset of the Cartesian product of the adhered domains; i.e. $D_1 \times \cdots \times D_m$. Denoted by R.

duration	price	channel_id
30	99.9 $	2
60	149.9 $	2
90	199.9 $	1

Subscription

$H_R = \{duration, price, channel\_id\}$

$R = \{< 30, 99.9\$, 2 >, < 60, 149.9\$, 2 >, < 90, 199.9\$, 1 >\}$

Programming Languages Definition

A Relation entails one Relation Variable (referred as H_R earlier), which has an arbitrary Relation Value (previously referred as R) similar to snapshots of a memory slot.

datetime	amount	wallet_id
1708731399.7138	10	13
1681673948.4366	83	4

Transaction

Degree. Number of the attributes; i.e. $|H_R|$.

Cardinality. Number of the tuples in the body.

e.g. The aforementioned *Transaction* relation has a degree of 3 and a cardinality of 2.

Relation Properties

- UNORDERED, regarding both the <u>Heading</u> and the <u>Body</u>.
  - The accurate notation for H_R: $\{< D_1 : A_1 >, \ldots, < D_m : A_m >\}$
  - The accurate notation for a tuple in the body: $\{< A_1 : a_1 >, \ldots, < A_m : a_m >\}$
- Distinctive regarding the tuples inside the body.
  - The most significant disparity between "Relational Algebra" and "SQL."
- Meaningful zero-degree relation.

- Extra: Domains
  - Emanate from Type Theory.
  - Not necessarily distinctive inside some specific relation; multiple attributes (semantics) over the same domain are plausible.

Where We Stand Now?

Relational Databases History

Relational Algebra (and DBs)

Edgar F. Codd, A Relational Model of Data for Large Shared Data Banks, 1970.

ER Diagram

Peter Chen, 1976.

Oracle's Commercial SQL

Based on IBM's work on SQL, 1979.

ER and Relational Algebra: Moving To and Fro

Entity (ER).

Moving To and Fro - Cont.

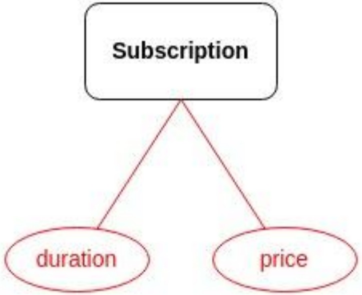
Relation (Algebra).

R1: Subscription

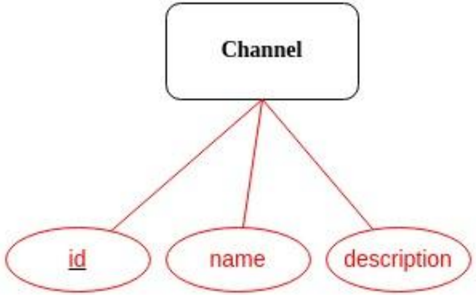
R2: Channel

Moving To and Fro - Cont.

Attribute (ER).



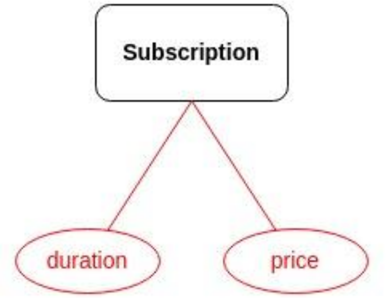
R1: Subscription

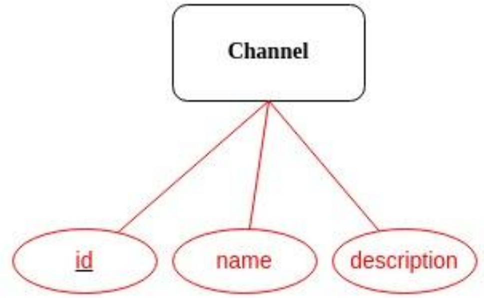


R2: Channel

Moving To and Fro - Cont.

Attribute (Algebra).





duration	price

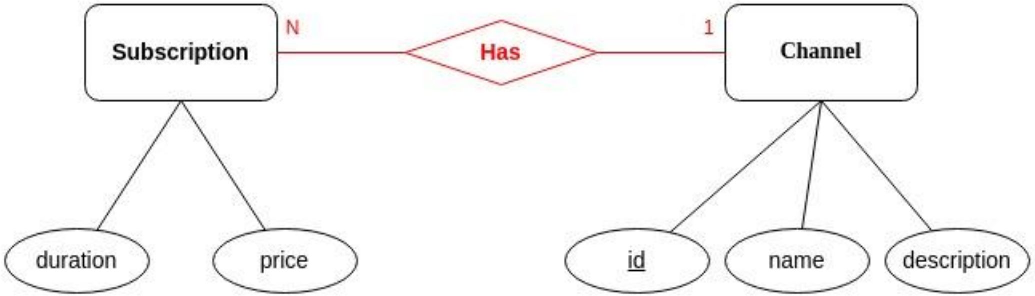
R1: Subscription

id	name	description

R2: Channel

Moving To and Fro - Cont.

Relationship (ER).



duration	price

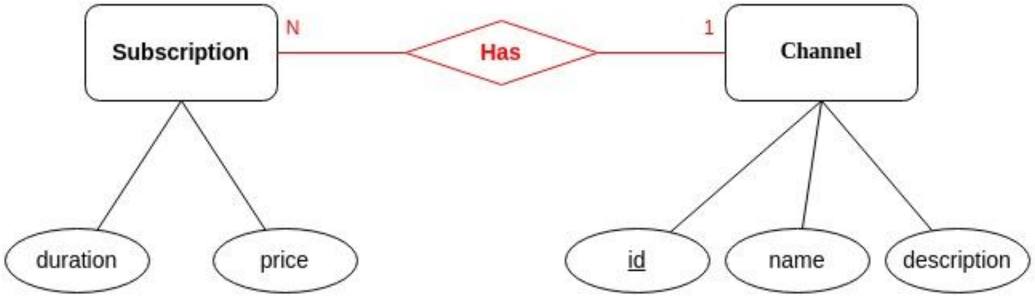
R1: Subscription

id	name	description

R2: Channel

Moving To and Fro - Cont.

Foreign Key (Algebra).



duration	price	channel_id

R1: Subscription

id	name	description

R2: Channel

Moving To and Fro - Extra

Tuples (Algebra).

?

duration	price	channel_id
7	10 $	9
30	35 $	9

R1: Subscription

id	name	description
9	“mySport”	“all about sports.”

R2: Channel

Relation - Manipulation

Operations

Closed Property. refers to the property that the result of a relational algebra operation, when applied to relations, will also be relations.

Operations.

1. Common Set-Theoretic
    a. Union
    b. Intersect
    c. Minus
    d. Times
2. Specific
    a. Project
    b. Restrict
    c. Join
    d. …

Common Set-Theoretic Operations

- All four Common Set-Theoretic operations are binary.
- Type/Union-Compatible of Two Operands
  - $H_{R_1} = H_{R_2}$

Union $\bigcup$ **Intersect** $\bigcap$ **Minus** $-$ **.** if R_1 and R_2 are two type-compatible relations and R_3 the result of an operation,

$$\forall op \in \{\cup \cap -\} : R_3 = R_1 op R_2$$

Times $\times$ **.** If R_1 and R_2 are two sets that $H_{R_1} \cap H_{R_2} = \emptyset$ holds between them, the result of a Cartesian product between them (R_3) is defined as follows:

- $H_{R_3} = H_{R_1} \cup H_{R_2}$
- $R_3 = R_1 \times R_2$

Common Set-Theoretic Operations - Cont.

Times $\times$**.** e.g.

datetime	amount	wallet_id
1708…	10	13
1681…	83	4

Transaction

id	balance
13	10
4	83
9	0

Wallet

datetime	amount	wallet_id	id	balance
1708…	10	13	13	10
1681…	83	4	13	10
1708…	10	13	4	83
1681…	83	4	4	83
1708…	10	13	9	0
1681…	83	4	9	0

Transaction x wallet

Common Set-Theoretic Operations - Cont.

Times $\times$**.** What's Wrong with this Cartesian product?

A	B	C
a1	b1	c1

$R_1 = \{< a_1, b_1, c_1 >\}$

X	Y
x1	y1
x2	y2

$R_2 = \{< x_1, y_1 >, < x_2, y_2 >\}$

A	B	C	X	Y
a1	b1	c1	x1	y1
a1	b1	c1	x2	y2

$R_3 = R_1 \times R_2 =?$

Specific Operations - Projection

- Notation: $\Pi_{<attrs>}(R)$
- Semantic: Returns a new relation of the same tuples, yet different heading corresponding to the demanded attributes in $.< attrs >$
- $< attrs >$ can be a list of integers rather than specific attribute names.
- A Unary operation (Why?).
- Problem: find list of all channel ids associated with channels which have no subscriptions.

duration	price	channel_id
7	10 $	9
30	35 $	9

Subscription

id	name	description
9	“mySport”	“all about sports.”
15	“CSOnline”	“a Computer Science channel”

Channel

$$R := \Pi_{<id>}(Channel) - \Pi_{<channel\_id>}(Subscription)$$

Specific Operations - Rename

- Notation: $\rho_{new/old}R$
- Semantic: Returns a new relation of the same tuples and same columns, yet the column named *old* is presently called *new* in the resultant relation.
- A Unary operation.
- Another prevalent notation: $\Pi_{(id\ RENAME\ AS\ studentId, stdNum, name\ RENAME\ AS\ firstName, lastName)}(R)$

Specific Operations - Restriction (Selection)

- Notation: $\sigma_c(R)$
- Semantic: Returns some horizontal subset of R in which each tuple satisfies the constraint c.
- A Unary operation.
- Problem: find all durations associated with subscriptions which cost no more than 30$ and are not pertinent to channel with the id = 9.

duration	price	channel_id
7	10 $	9
30	35 $	9
10	20 $	23

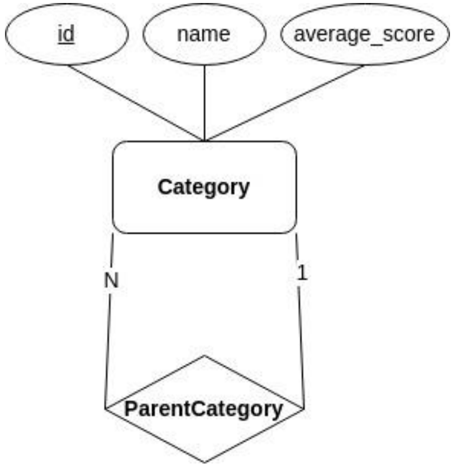
Subscription

$$\Pi_{duration}(\sigma_{(price \leq 30) \wedge (channel\_id \neq 9)}(Subscription))$$

Specific Operations - Restriction - Cont.

- Commutativity Property: $\sigma_{c_1}(\sigma_{c_2}(R)) = \sigma_{c_2}(\sigma_{c_1}(R))$.
- $\sigma_{c_1 \wedge c2}(R) = \sigma_{c_1}(R) \cap \sigma_{c_2}(R)$
- $\sigma_{c_1 \vee c2}(R) = \sigma_{c_1}(R) \cup \sigma_{c_2}(R)$
- $\sigma_{\neg c}(R) = R - \sigma_c(R)$

Problem: In the following subsystem, return list of all category ids that have an average score more than 4 (out of 5) or have an immediate child with such an average score.

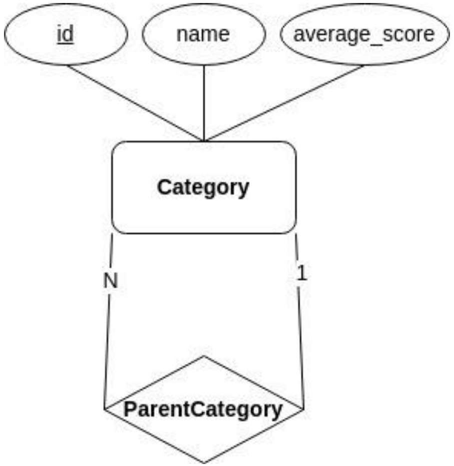


Specific Operations - Restriction - Cont.

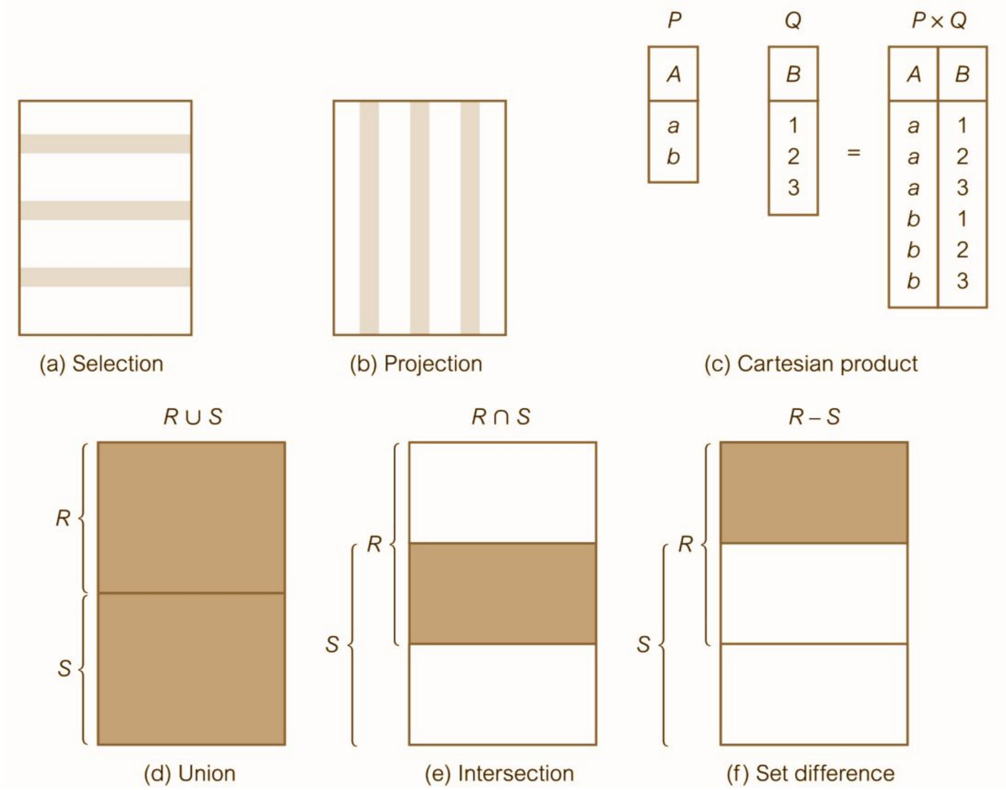
- Commutativity Property: $\sigma_{c_1}(\sigma_{c_2}(R)) = \sigma_{c_2}(\sigma_{c_1}(R))$.
- $\sigma_{c_1 \wedge c2}(R) = \sigma_{c_1}(R) \cap \sigma_{c_2}(R)$
- $\sigma_{c_1 \vee c2}(R) = \sigma_{c_1}(R) \cup \sigma_{c_2}(R)$
- $\sigma_{\neg c}(R) = R - \sigma_c(R)$

Problem: In the following subsystem, return list of all category ids that have an average score more than 4 (out of 5) or have an immediate child with such an average score.

$$\Pi_{id}(\sigma_{average\_score>4}(C) \cup$$
$$\Pi_{id}(\sigma_{id=pId \wedge child\_avg>4}($$
$$C \times \Pi_{parent\_id\ RENAME\ AS\ pId, average\_score\ RENAME\ AS\ child\_avg}(C)))$$



Operations So Far.



Keys

Super Key and Candidate Key

Super Key

Any subset of H_R that has the uniqueness trait and thus is utilized to identify a tuple.

Given N as the number of all super keys in some relation with m attributes, the following inequality holds:

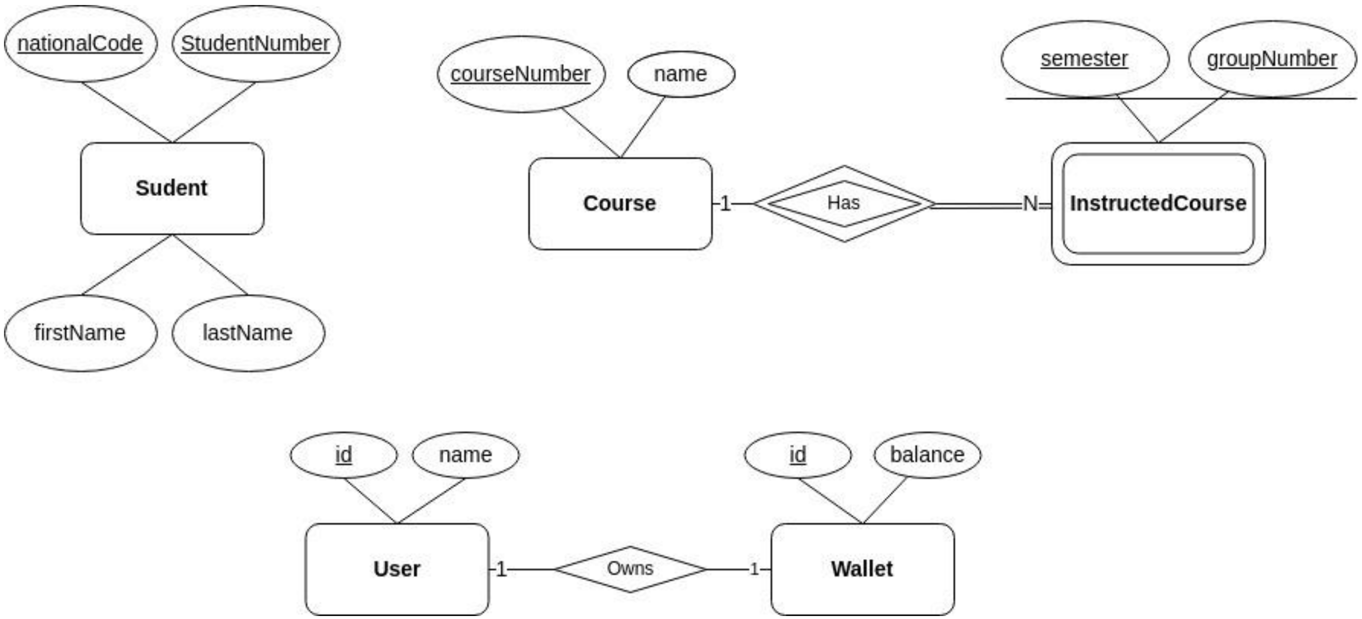
$$1 \leq N \leq 2^m - 1$$

Candidate Key

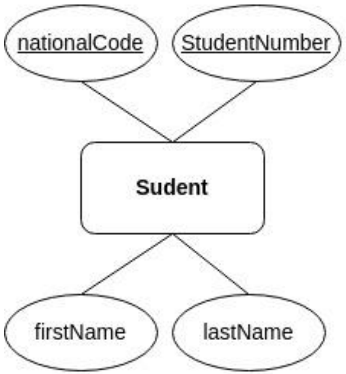
A super key possessing the minimality attribute; i.e. none of its subset is considered a super key.

Super Key and Candidate Key - Cont.

Problem: Find candidate keys in the below ER diagrams.

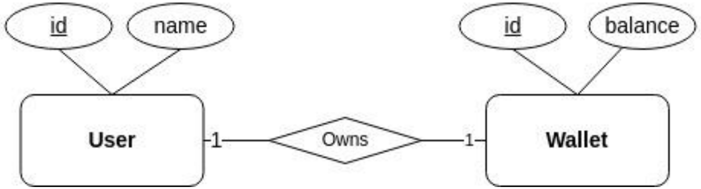


Super Key and Candidate Key - Cont.



$$Student(nationalCode,\ studentNumber,\ firstName,\ lastName)$$
$$\rightarrow CKs = \{\{nationalCode\},\ \{studentNumber\}\}$$

Super Key and Candidate Key - Cont.



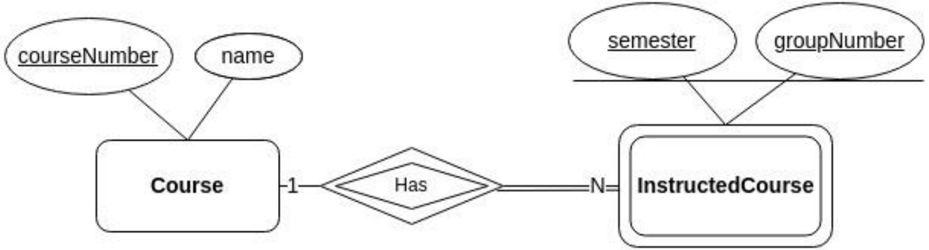
$User(id, name)$

$User(id, balance, user\_id)$

$\rightarrow CKs_{User} = \{\{id\}\}$

$\rightarrow CKs_{Wallet} = \{\{id\},\ \{user\_id\}\}$

Super Key and Candidate Key - Cont.



$Course(courseNumber, name)$

$InstructedCourse(semester,\ groupNumber,\ courseNumber,\ daysOfWeek,\ startingTime)$

$\rightarrow CKs_{Course} = \{\{courseNumber\}\}$

$\rightarrow CKs_{InstructedCourse} = \{\{semester,\ groupNumber,\ courseNumber\}\}$

Primary Key

An augmented semantic to one and only candidate keys that brings no more theoretical explanation to the table whatsoever.

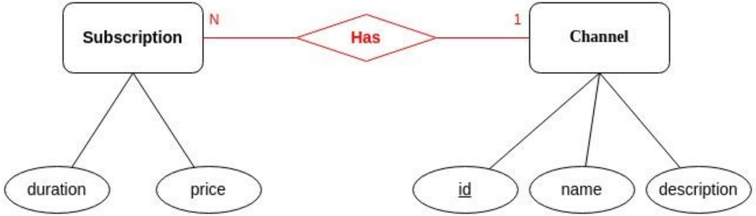
A design decision that has practical justifications to exist:

- The most prevailing identifier in the business domain.
- Its value for some tuple almost never gets changed.
- 1NF (Normalization).
- Automatic Indexes (In SQL).

Foreign Key

An attribute (or a set of attributes) in *R2* is called a foreign key if it corresponds to a another attribute (or set of attributes) with the same domain(s) in *R1*.

Represents a relationship (e.g. ER's relationships) between two relations in Relational Algebra.



duration	price	channel_id

R1: Subscription

id	name	description

R2: Channel

Foreign keys are the explicit statement of a relationship. However, any common attribute (identical in domains) can act as a relationship.

Relation - Manipulation - Cont.

Specific Operations - Theta Join

- Notation: $R_1 \bowtie_c R_2$
- Semantic: Joins two relations based on some defined relationship (mostly foreign keys).
- A binary operation.
- Equivalent to a cartesian product ($\times$) followed by a restriction operator (σ_c):
  - $R_1 \bowtie_c R_2 = \sigma_c(R_1 \times R_2)$
- e.g.

duration	price	channel_id
7	10 $	9
30	35 $	9
10	20 $	15

$Subscription$

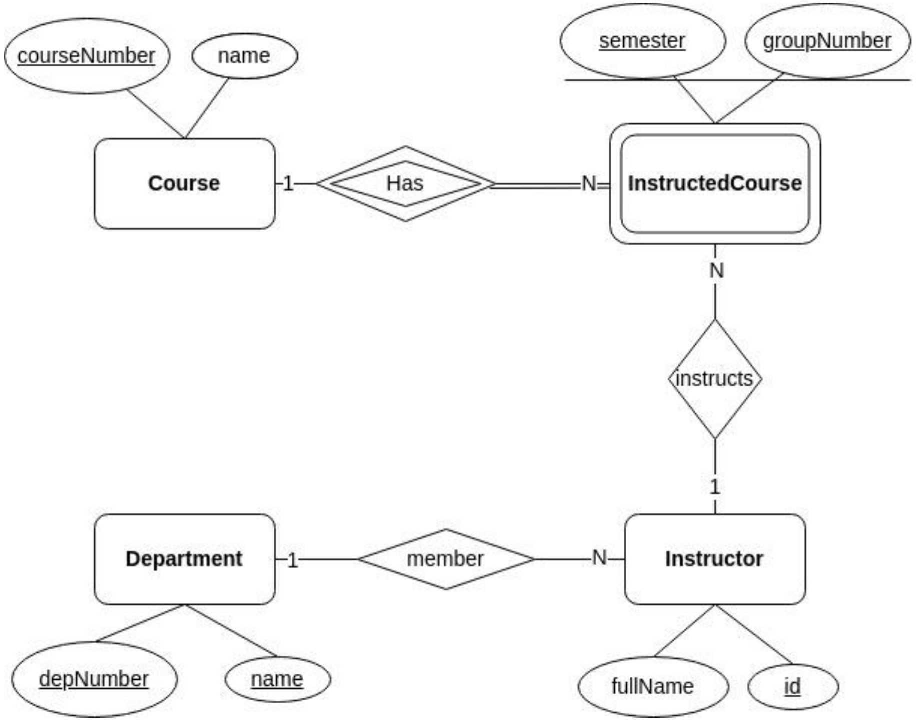
id	name
9	“mySport”
15	“CSOnline”

$Channel$

id	name	duration	price	channel_id
9	“mySport”	7	10 $	9
9	“mySport”	30	35 $	9
15	“CSOnline”	10	20 $	15

$Channel \bowtie_{(Channel.id=Subscription.channel\_id)} Subscription$

Specific Operations - Theta Join - Cont.



$Course(courseNumber, name)$

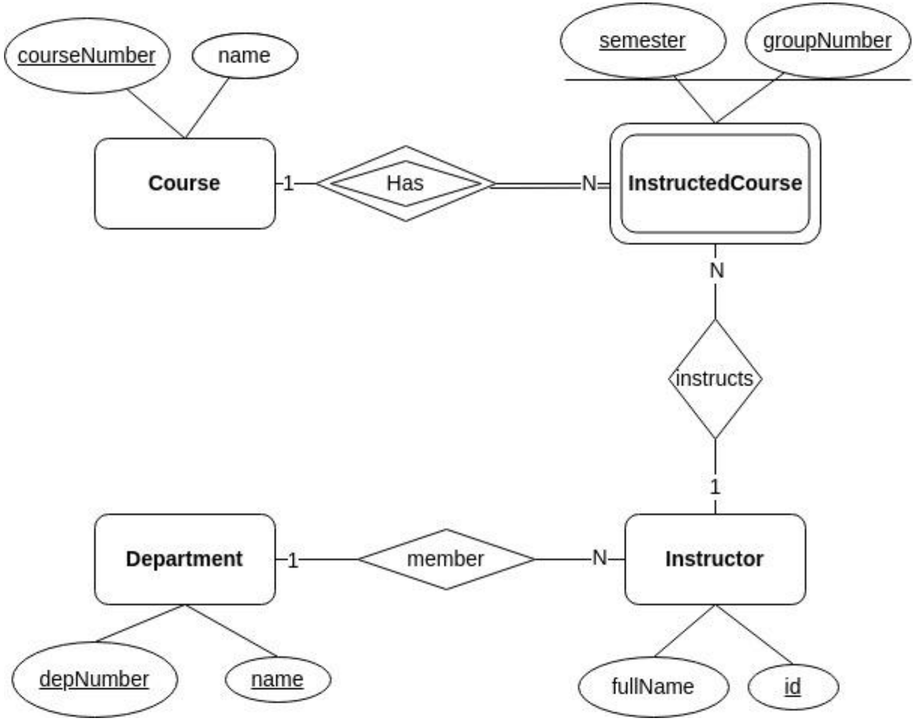
$InstructedCourse(course, semester, groupNumber, instructor)$

$Instructor(id, fullName, memberDepNumber)$

$Department(depNumber, name)$

Problem: What were the group numbers of all the Database Design courses that were instructed by Mahdi Akhi in the last semester in the CE department?

Specific Operations - Theta Join - Cont.



Problem: What were the group numbers of all the Database Design courses that were instructed by Mahdi Akhi in the last semester in the CE department?

$$\Pi_{groupNumber}(\\ \sigma_{D.depName='CE'}(\\ \rho_{depName/name}(D) \bowtie_{D.depNumber=I.department}\\ \sigma_{I.fullName='Mahdi\ Akhi'}(\\ I \bowtie_{I.id=IC.instructor}\\ \sigma_{IC.semester=14021}(\\ IC \bowtie_{IC.course=C.courseNumber} C))))$$

Specific Operations - Natural Join

- Notation: $R_1 \bowtie R_2$
- Semantic: Joins two relations based on some implied relationship; joins over all common attributes.
- A binary operation.
- e.g.

datetime	userId	msg	bId
170…	23	‘foo’	3
169…	48	‘bar’	3
163…	61	‘test’	10
171…	23	‘fooo’	10

$Ticket$

bId	name
3	‘Varzesh4’
10	‘Tap40’

$Business$

bId	name	datetime	userId	msg
3	‘Varzesh4’	170…	23	‘foo’
3	‘Varzesh4’	169…	48	‘bar’
10	‘Tap40’	163…	61	‘test’
10	‘Tap40’	171…	23	‘fooo’

$Business \bowtie Ticket$

Relation - Manipulation - Extended Operations

Specific Operations - Semi Join

- Notation: $R_1 \ltimes_C R_2$
- Semantic: Resultant tuples of a theta join over two relations, considering only the attributes of the operand on the left hand-side.
- Equivalent to: $R_1 \ltimes_C R_2 = \Pi_{<H_{R_1}>}(R_1 \bowtie_C R_2)$
- e.g.

duration	price	channel_id
7	10 $	9
30	35 $	9
10	20 $	15

$Subscription$

id	name
9	'mySport'
12	'BitNow'
15	'CSOnline

$Channel$

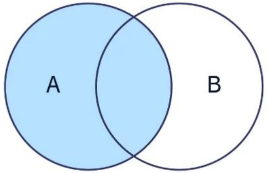
id	name
9	"mySport"
15	"CSOnline"

$Channel \ltimes_{Channel.id=Subscription.channel\_id} Subscription$

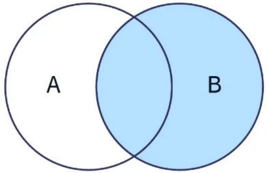
Specific Operations - Outer Joins

Left (Right) Outer Join returns all the tuples in the relation on the left (right) hand-side of the operation even if there was no match found for some of them in the required Join. In a similar vein, Full Outer Join returns all the tuples in both mentioned relations.

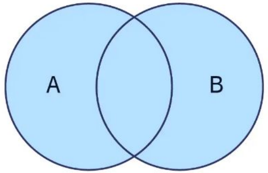
- ❖ Left Outer Join: $R_1 ⟕_C R_2$



- ❖ Right Outer Join: $R_1 ⟖_C R_2$



- ❖ Full Outer Join: $R_1 ⟗_C R_2$



Specific Operations - Outer Joins - Cont.

An illustration of all possible types of Outer Join

orders_id	user_id
o1	c1
o2	c2
o3	null

id
c1
c2
c3

orders_id	customer_id	id	
o1	c1	c1	*tuple1*
o2	c2	c2	*tuple2*
o3	null	null	*tuple3*
null	null	c3	*tuple4*

$$Orders ⟕_{User.id=Orders.user\_id} User = \{tuple1, tuple2, tuple3\}$$

$$Orders ⟖_{User.id=Orders.user\_id} User = \{tuple1, tuple2, tuple4\}$$

$$Orders ⟗_{User.id=Orders.user\_id} User = \{tuple1, tuple2, tuple3, tuple4\}$$

Relational Algebra - Problems

Problems

1. Stock Market. Take the following Stock Market system into consideration. Return the fields of the organizations which have none of their symbols bought by the shareholders with ‘Saman’ bank account.

$$ShareHolder(id, fullName)$$

$$BankAccount(accNumber, shareholderId, bankName)$$

$$Symbol(id, name, orgId)$$

$$Organization\ (id, name, field)$$

$$buy(amount, price, shareholderId, symId)$$

Problems

1. Stock Market.

$$\Pi_{field}(Org) -$$
$$\Pi_{field}[(\rho_{orgId/id}(Org) \bowtie Sym) \ltimes_{Sym.id=Buy.symId}$$
$$\sigma_{bankName='Saman'}(Buy \bowtie_{Buy.shareholderId=SH.id} SH \bowtie_{BA.shareholderId=SH.id} BA)]$$

Problems

2. Instagram Advertising. Consider the following advertising system in which there are *PageOwner*s who place ads on their Instagram *Page*s. Owners define *AdPlan*s for each *Page*. A plan represents the *duration* an advertisement stays before it is removed and the *price* for the advertising. In our system, we designate a *Support* member for each *PageOwner* to assist them in using the system. Find all the *PageOwner id*s who have pages that are not private, with *AdPlan*s priced at more than $20, and that have already had corresponding placed *Ad*s. Find the name of the *Support* member associated with each of these *PageOwner*s in the same query (if applicable).

Problems

2. Instagram Advertising.

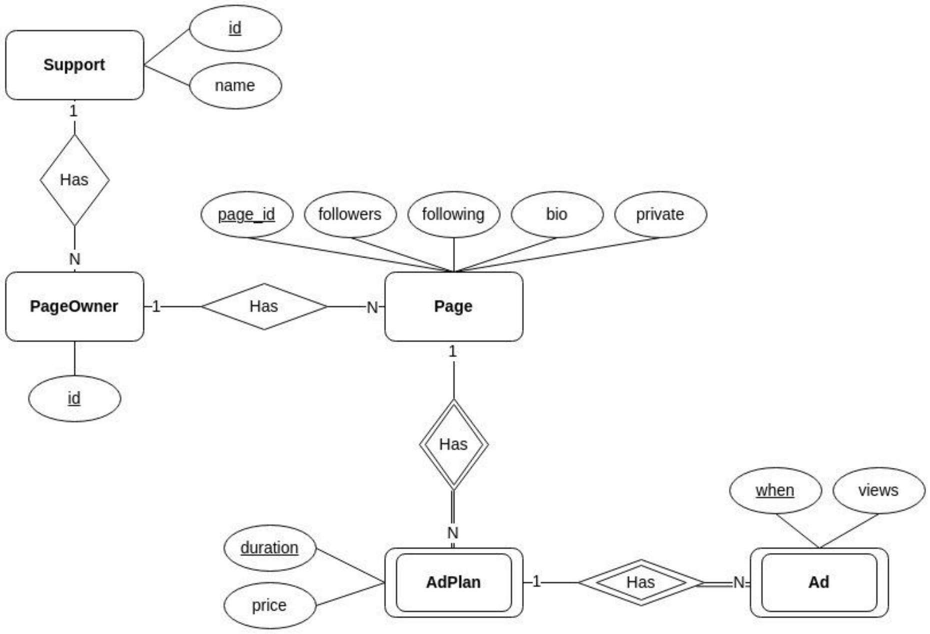
$PageOwner(id,\ support\_id)$

$Page(page\_id, followers, following, bio, private, owner\_id)$

$AdPlan(page\_id, duration, price)$

$Ad(page\_id, duration, when\_placed, views)$

$Support(id, name)$



Problems

2. Instagram Advertising.

$$\Pi_{<id,\, name>}[$$

$$PageOwner \ltimes_{PageOwner.id=Page.owner\_id}($$

$$\sigma_{private=false \,\wedge\, price>20 \,\wedge\, when\_placed<NOW}(Page \bowtie AdPlan \bowtie Ad))$$

$$\bowtie \rho_{support\_id/id}(Support)]$$

End.