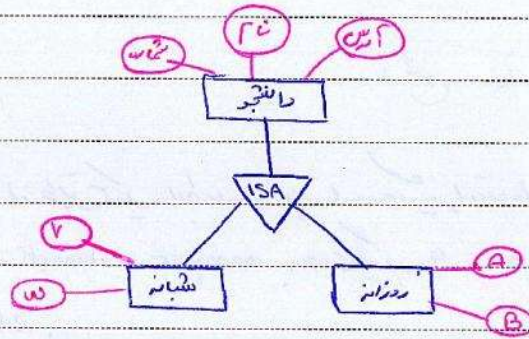
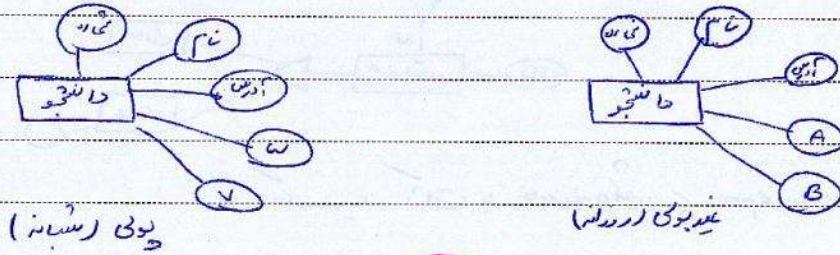


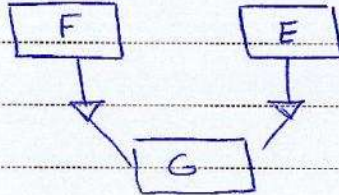
۱۰. تعمیم این تکنیک تعمیم (generalization) می باشد  
 تعمیم به این صورت است که از یک نوع خاص به یک نوع کلی تر رسیدن است (مثلاً در یک دسته بندی از یک نوع خاص به یک نوع کلی تر رسیدن است)  
 به صورتی که هر چه خاص تر باشد به آن دسته بندی در یک نوع کلی تر رسیدن است.



تعمیم دو طرفه است: ۱. مثال‌ها در یک نوع خاص به یک نوع کلی تر رسیدن است.  
 ۲. آن نوع کلی تر به یک نوع خاص رسیدن است.

این نوعی از تعمیم در یک نوع خاص به یک نوع کلی تر رسیدن است (Category)  
 به این معنی که هر چه خاص تر باشد به آن دسته بندی در یک نوع کلی تر رسیدن است.  
 به صورتی که هر چه خاص تر باشد به آن دسته بندی در یک نوع کلی تر رسیدن است.

g : مجموعه‌های E و F  
 E ...  
 F ...

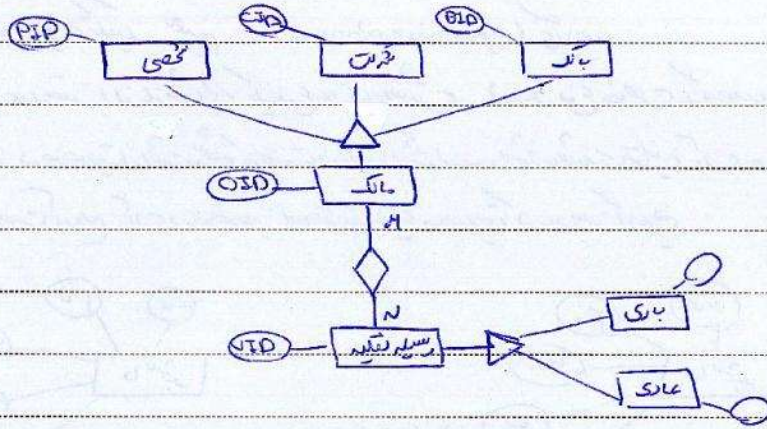


$$g \subseteq E \cup F$$

مجموعه‌های E و F در یک نوع خاص به یک نوع کلی تر رسیدن است.  
 به این معنی که هر چه خاص تر باشد به آن دسته بندی در یک نوع کلی تر رسیدن است.

Subject:

Year:      Month:      Date: ( )



بانک به شرکت یا هر شخصی که هم یکی از آنها

### Aggregation (تجميع)

3 شرط دسته

یک مجموعه روابط که ارتباط دارند از طریق یک رابطه به صورتی که اینها ارتباط داشته باشند  
 مثلا در سطح job, employee, branch, manager به صورتی که 4 تا با هم ارتباط داشته باشند  
 3 تا به هم ربطی ندارند اما با هم دارند

برای اینکه اینها با هم ارتباط داشته باشند درجه اولی به نام تجميع (تجميع) aggregation استفاده می کنند

تجميع عبارتست از دو یا چند نوع موجودیت، شرکت کننده در این ارتباط به صورتی که نوع موجودیت اشتراکی که همه اینها را در بر می گیرد به صورتی که مرکب

Subject:

Year. Month. Date. ( )

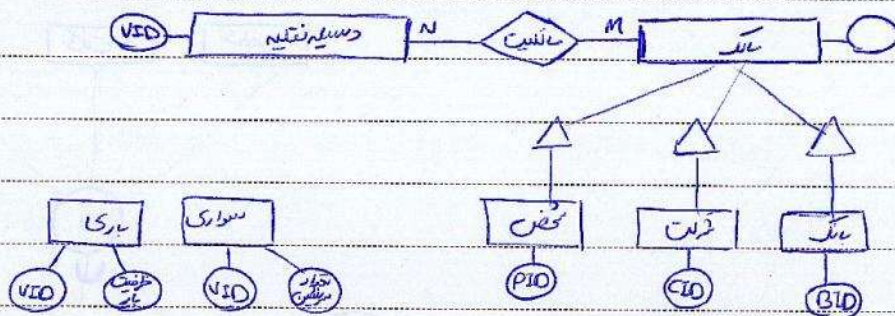
7.15 علمه حقیق

DB Exercise #1

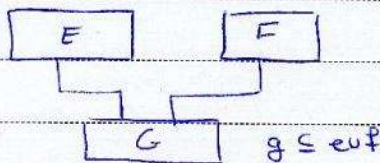
Send to: dbhw87@gmail.com

Deadline: next monday

Group: groups.google.com/group/db.87881



ساز به این نوع طراحی می باشد که از U-type یا کسپین به هم استفاده کرد که در طراحی گوناگونی است که قابلیت با  
 خود نظر داشته در این مثال می توان از کسپین استفاده کرد که در این database به زبان برنامه ریزی به هم پیوسته  
 سازی و استفاده به هم پیوسته که از U-type استفاده کرد  
 چون این به هم پیوسته و کسپین و در این نوع طراحی را استفاده می کنند  
 زیر نوع مالاتر از این نوع



توزیع union-type و کسپین به هم

کسپین به هم

Union-type

- 1- زیر نوع چنانچه چندین وجودی از نوع دارد
  - 2- زیر نوع چنانچه وجودی از نوع دارد
  - 3- کسپین چنانچه از نوع به هم پیوسته ای از اجتماع وجودی
- ماتریس ساز

Subject:

Year. Month. Date. ( )

تفاوت های میان U-type و زیر نوع مسترک (مربوط به وجود دارد)

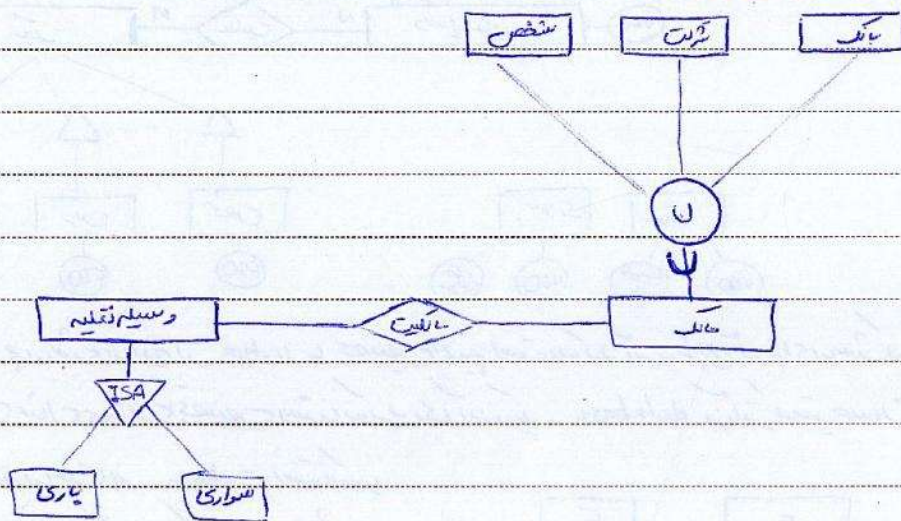
میگنونه در زیر نوع مسترک باید در همه زیر نوع ها وجود داشته باشند

یک نمونه در U-type باید فقط در یک زیر نوع وجود داشته باشند

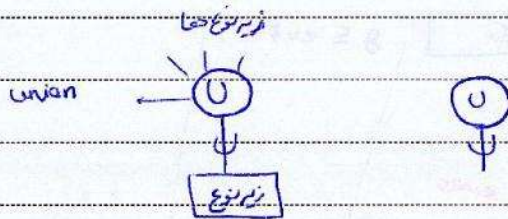
- نمونه های موجود در زیر نوع مسترک تمام خصوصیات زیر نوع ها را ارث می برد

- نمونه های موجود در U-type تمام خصوصیات زیر نوع را ارث می برد

شکل اصلی سوال در صورت



فرم های زیر نوع ها



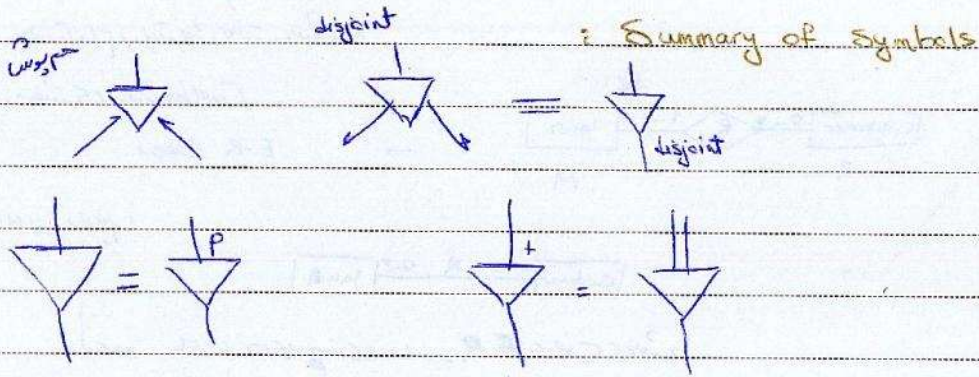
Aggregation

در aggregation با استفاده از خط لبه و خصوصیات های اتصال یک یا دو خصوصیت اجزای در دست گرفتن و تبدیل شدن

موجودیت یک رابطه داشته باشند مثلا در این مثال به عنوان یک مدیر و یک کارمند و manager رابطه ای جداگانه داشته

باشند و استفاده از job، person، branch، و رابطه work-on. یک خصوصیت اجزای تبدیل شدن

فقط توسط یک رابطه با manager در ارتباط است.



### UML

هدف از آن است که برای مدل‌سازی سیستم‌ها (data representation) در یک نرم‌افزار استفاده شود.  
 ERD تمایزی از مدل‌سازی سیستم‌ها است.  
 به وقت data را می‌توانیم در یک سیستم نمایش دهیم. و این سیستم را می‌توانیم در UML نمایش دهیم. برای این منظور در UML نمایش می‌دهیم که این سیستم نرم‌افزاری است.

class diagram: سیستم ERO است.

USE case: بیان کاربر سیستم را نشان می‌دهد. هر عملی که توسط user انجام می‌دهد و نتیجه آن در سیستم جواب داده شود.

Activity: بیان نحوه انجام کارهای مختلف در سیستم را نشان می‌دهد.

Implementation: اجرای سیستم در یک محیط خاص را نشان می‌دهد.

UML (point): (object) بیان می‌کند که در ERD آنچه که مدل‌سازی شده است چه چیزهایی هستند.

UML (class): (object) بیان می‌کند که در ERD چه چیزهایی را بیان می‌کند.  
 یک شیء می‌تواند به صورت یک شیء خاص در داده‌ها یا به صورت یک شیء خاص در داده‌ها باشد.

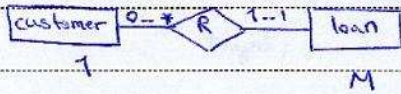
روابط غیر عددی را می‌توانیم در UML نمایش داده می‌شوند.  
 به سبب آن که ابتدا باید به صورت عددی تقسیم کرد.

Subject:

Year. Month. Date. ( )

B

یک مشتری می تواند یک یا چند وام داشته باشد  
یک وام یک مشتری را می تواند داشته باشد



رابطه E-R

رابطه UML داریم:



رابطه UML و ER یکی هستند

جلسه هفتم 7, 20

### E-R Design Decisions

1. استفاده از موجودیت‌ها یا صفت‌ها  
 صفت‌ها: اگر مفهوم دارای چند مفهوم منحصر به فرد باشد، یا اینکه چندین بار در اصل (رابطه یا موجودیت‌ها) یکبار ظاهر شود.  
 صفت: در غیر این صورت

2. استفاده از موجودیت یا رابطه:

اگر مفهوم در دنیای واقعی موجودیت است، غیر از موجودیت مختصر برای مفهوم یا صفت این را

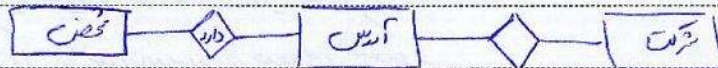
3. تجربه رابطه‌های صفتی: صفتی (n:m)

4. قرار دادن صفت یا رابطه

5. استفاده از مفهوم و تقسیم و یا صفت: حجم، اگر مفهوم منحصر به فرد است، دارای صفت‌ها یا صفت‌ها یا اینکه موجودیت‌ها را با خاصیت را اعمال کنند.

صفت، در غیر این صورت

6. اگر صفت سلباً یا موجودیت: صفت سلباً، اگر مفهوم صفت دارای صفت معنای منحصر به فرد است.  
 موجودیت، اگر مفهوم صفت به ازای استفاده شده باشد، این اطلاعات باید به صورتی که نوشته شده است.



Subject:

Year. Month. Date. ( )

α

7- استفاده از جدولیت در سیستم‌های داده‌ای

8- استفاده از تجمیع (Aggregation) در طراحی سیستم‌های داده‌ای

### طراحی Database در سیستم‌های داده‌ای

Requirement Analysis

تجزیه و تحلیل نیازها و مشخصات سیستم

entity: conceptual and logic Design

طراحی مفهومی و منطقی از اجزای سیستم

Implementation: تبدیل طراحی منطقی به سیستم عملی

اجرای سیستم و پیاده‌سازی

validation and Evaluation

بررسی و ارزیابی سیستم

operation & maintenance

عملیات و نگهداری سیستم

interface کاربر: طراحی رابط کاربری و سیستم

### System Design

طراحی سیستم: تبدیل نیازها به طراحی مفهومی

ER model: مدل رابطه‌ای

ER model: تبدیل مدل رابطه‌ای به مدل فیزیکی

### Implementation

اجرای سیستم: پیاده‌سازی در Oracle

Subject:

Year. Month. Date. ( )

Evolution

مستند طراحی شده ایچهای است که مشتری تغییر کرده اگر لازم باشد به طراحی تغییردهیم

DataBase Design For Banking Enterprise

بانی داریم به چندین شعبه در کشور و نیاز

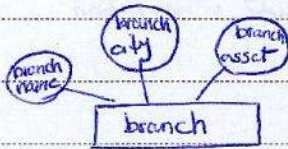
مشتری به شماره مشتری و اسم و نام و آدرس

حسابهای مختلف account داشته باشد و به شماره در ارتباط باشد و حسابهای

اطلاعات باشد به رابطه چندین مشتری account وجود دارد

اولین کارشناسی حسابها است

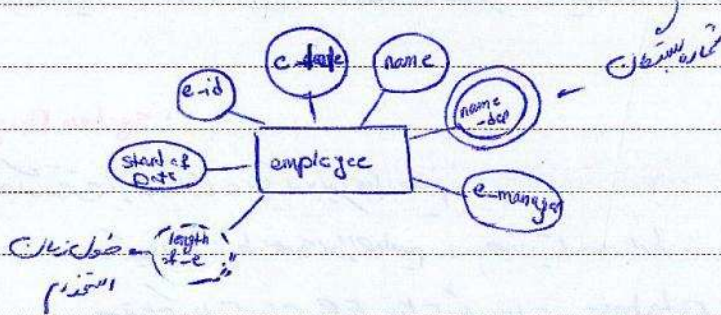
1. یک بانک چندین شعبه دارد که هر شعبه یک سری حسابات دارد



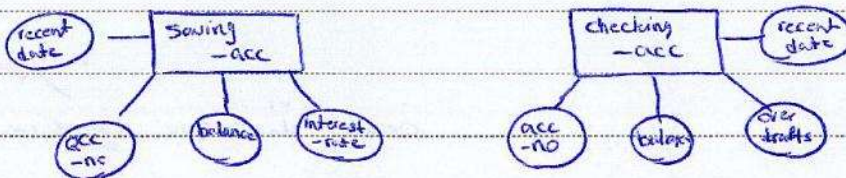
2. یک مشتری داریم:



3. کارکنان:



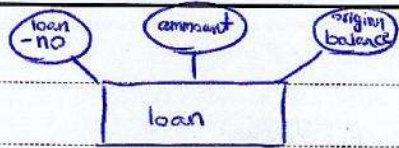
4. account:



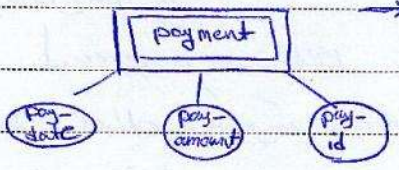


Subject: \_\_\_\_\_

Year.    Month.    Date.    ( )



ifb-5



مستطابق payment अनुसार

تاریخ کے مطابق

Subject: \_\_\_\_\_

Year. \_\_\_\_\_ Month. \_\_\_\_\_ Date. ( ) \_\_\_\_\_

7, 22, 2022

### Reduction of an E-R Schema to Tables

وقتی که یک E-R اسکیم را می بینیم، چطور می توانیم آن را به جدول تبدیل کنیم؟

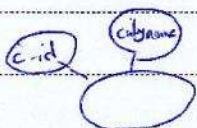
به برای این کار جدول می سازیم

در جدول هر یک از اشیاء در یک جدول و مشخص می کند رابطه اش در Database

فرد اول برای این کار DB رابطه ای، استخراج جدول از مدل E-R است

جدول ها و صفات ها

هر دو ویژگی یک ویژگی صفت دارد و صفت با نام به صورتی که جدول را صفت است



در صفت به اجزای می شود تا بتواند رابطه صفت را بداند

در صفت می تواند به صفت را از اجزای می شود تا بتواند رابطه صفت را بداند

در صفت می تواند به صفت را از اجزای می شود تا بتواند رابطه صفت را بداند

در صفت چندتا می تواند به صفت را از اجزای می شود تا بتواند رابطه صفت را بداند

با استفاده از جدول می توانیم رابطه ای را مشخص کنیم

### Representing Weak Entity Set

اگر یک weak entity set داشته باشیم، به یک موجودیت قوی در رابطه می شود

جمله موجودیت قوی را در موجودیت ضعیف قرار می دهیم، جمله اصلی در موجودیت ضعیف را به جمله اصلی قوی می دهیم

و جمله در موجودیت ضعیف، جمله اصلی در موجودیت قوی می شود

loan number + payment number ← side

صفت می تواند به صفت ضعیف (یا جمله موجودیت قوی) → جمله اصلی جدول Entity

### Representing Relationship sets as Tables

چگونه می توانیم رابطه را به جدول تبدیل کنیم؟

اگر رابطه را چند چند می بینیم، می توانیم هر یک از اجزای اصلی موجودیت اول را به جمله اصلی موجودیت دوم قرار می دهیم

و هر صفت که رابطه دارد می تواند به جمله اول می توانیم



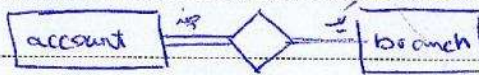
Subject:

Year.      Month.      Date.      ( )

### Redundancy of Tables

acc. no	balance	branch name	bc	assets
a-1	200			
a-2	2050			

اگر رابطه چند به یک و یک به یک باشد می توانیم صحت های این برای  
رابطه جدول برای حساب و جدول account را  
رابطه یک به چند است پس دو باره می توانیم  
جدول را در دو تابع entity برای branch  
ی پریم و تفصیلاتی آن را در جدول account

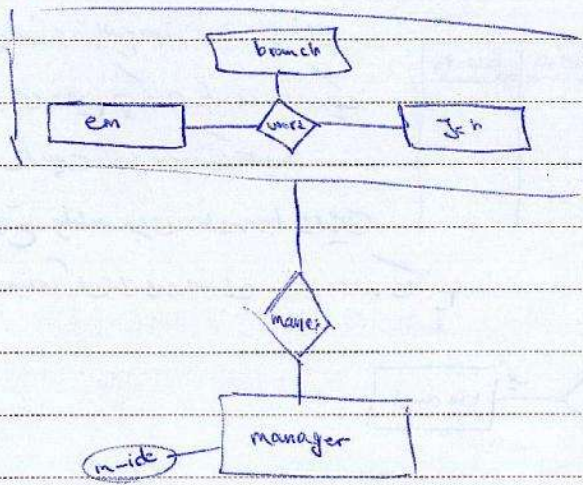


### Representing Specialization as Tables

Subject: \_\_\_\_\_

Year.    Month.    Date.    ( )

### Aggregation



Subject:

Year. Month. Date. ( )

جلسه دوم: 26, 7, 7

### Chapter 3 (Relational Model)

یک فضای رابطه‌ای به پیش زمینه‌ای برای بین رابطه‌ای رابطه‌ای

بین فضای رابطه‌ای به مجموعه‌ای از رابطه‌ها که در یک جدول مشخص شده است

به سبب و صفتها

به سبب و صفتها

$$R = (D_1, D_2, \dots, D_n)$$

صفتها

$$D_1 \times D_2 \times \dots \times D_n = (a_1, a_2, \dots, a_n)$$

$$a_i \in D_i$$

در یک رابطه، هر رابطه‌ای به یک مجموعه از سببها و صفتها

$D_1 = \text{Customer-name}$

صفتها

$D_2 = \text{street}$

$\{ \text{name, street, city} \}$

$D_3 = \text{city}$

### Attribute type

صفتها به نامها و صفتها

به مجموعه صفتها

در یک رابطه، هر رابطه‌ای صفتها به یک مجموعه از سببها و صفتها

به صفتها و سببها

در این رابطه، هر رابطه‌ای به یک مجموعه از سببها و صفتها

### Relation schema

Schema به مجموعه‌ای از سببها و صفتها

در واقع، هر رابطه‌ای که در یک رابطه

$R$  به مجموعه‌ای از instanceها که یک رابطه‌ای هستند

$R$  Type definition به مجموعه‌ای از S.variableها

هرگونه - سببها و صفتها

Subject:

Year. Month. Date. ( )

تجزیه tuple تجزیه ای است.

رابطه های آنرا تجزیه می کنند و هر چه در آن است را می بینیم. اینها هم نسبت به جای جدولها این تجزیه نسبت به جای سطرها تا آنجا تغییر نسبت

### Database

درون رابطه ای به یک اطلاعاتی می آید از رابطه ها

طراحی یک جدول اطلاعاتی برای سازمان - به تمام های کوچک می بینیم و هر وقت که یک جدول نشان می دهیم. می توانیم نسبت اطلاعات را از رابطه با هم و هم اطلاعات را یک جدول می بینیم که امنیت نداریم. اطلاعات از دست

دقیق برای جدولها Schema را تجزیه می کنیم همان برای یک اطلاعاتی است.

Schema تجزیه می شود یک اطلاعاتی

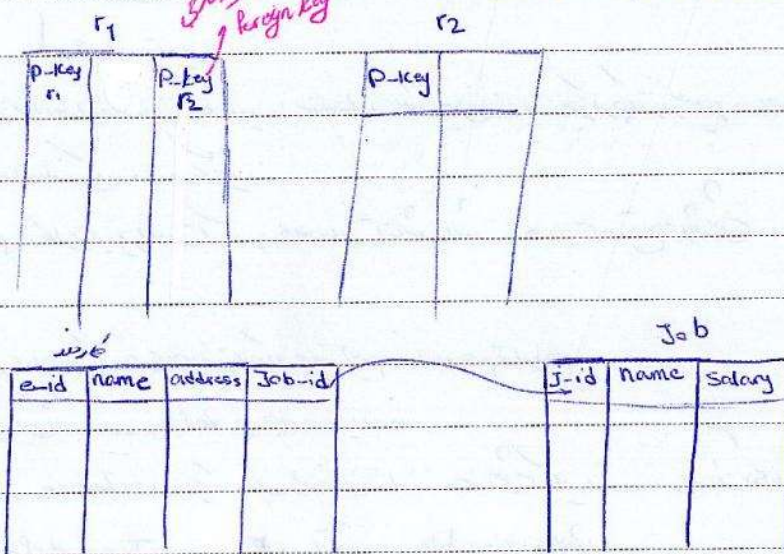
Instance جدول در یک جدول

جدولها در یک رابطه را انتخاب می کنیم. طوری که می توانیم

### Keys:

کلید اصلی می باشد که در یک سطر می آید. Instance از جدول شماره می کنند

Superkey: کلیدی که یک اطلاعات (سازمان) شماره می کنند



Subject:

Year. Month. Date. ( )

strong entity set هر یک مجزای با خصوصیت تکی به صله اصلی می باشد و صله اصلی به خصوصیت تکی

weak entity set اگر به خصوصیت ضعیف را نشانه کنیم به صله اصلی به خصوصیت تکی و صفت تکی را نشانه

Relationship set هر یک صله های اصلی به خصوصیت است

### Query Language

Query lang. استخراج اطلاعات از یک پایگاه اطلاعاتی را با یک زبان خاص Query می گویند  
Query به معنای پرسش است و در اطلاعات خاصی را می خواهم و آنرا Table و کشف اطلاعات را به خصوصیت می گویند  
که پرسشهای به تفویض کنیم چه اطلاعاتی را می خواهیم

### Relational Algebra

زبان ریاضی که بکس کالوس چه اطلاعاتی می خواهم و آنرا چگونه استخراج کند  
: operation

انتخاب کردن، تغییر کردن، اجتماع کردن، حذف کردن، تعریف  
هر یک از دستورات رابطه می تواند در خودی یک رابطه را خود را حذف کند

### Select Operation - Example

یک کلاس انتخاب می کنیم  $\sigma_{A=B \wedge D > 5}$  در مقایسه  $A=B$   
در استخراج این جابجایی را باید بکنیم حاصل آن در operation در نتیجه انتخاب می کنیم

$\sigma_p(r)$

به گزاره ای که شرایط می تواند  
ماتریس می شود

$\rightarrow$  بین گزاره ها

معنی  $\neg$   $\vee$   $\wedge$   $\rightarrow$   $\leftrightarrow$   $\neq$   $=$   $>$   $<$   $\geq$   $\leq$

Subject:

Year. Month. Date. ( )

### Project Operation

$(A, C) \cap B$

رابطه  $A$  و  $C$  را در  $B$  قرار می دهیم  
یعنی فقط قسمت های  $A$  و  $C$  را نشان می دهیم  
یعنی قسمت  $B$  را نشان می دهیم و قسمت های دیگر را

$(A \cup B) \cap C$

پیدا کردن تقاطع دو مجموعه  $A$  و  $B$  را داریم:

بعد از آنجا که این مجموعه ها را پیدا کردیم در  $C$  قرار می دهیم

### Union Operation

$A \cup B$

مجموعه  $A$  و  $B$  را با هم می یابیم  
یعنی هر دو مجموعه را با هم می یابیم  
و در  $C$  قرار می دهیم  
یعنی هر دو مجموعه را با هم می یابیم

### Difference operation

$A - B$

مجموعه  $A$  را در  $B$  قرار می دهیم  
یعنی رابطه  $A$  را در  $B$  قرار می دهیم  
مجموعه  $A$  را در  $B$  قرار می دهیم

### Cartesian Operation:

$A \times B$

مجموعه  $A$  و  $B$  را با هم می یابیم  
یعنی هر دو مجموعه را با هم می یابیم  
مجموعه  $A$  و  $B$  را با هم می یابیم

$A \times B = \{ (a, b) \mid a \in A \text{ and } b \in B \}$

$A \times B = B \times A$  و فقط برای مجموعه های خاص می شود.

در صورتی که هر دو مجموعه  $A$  و  $B$  را با هم می یابیم  
یعنی هر دو مجموعه را با هم می یابیم  
در صورتی که هر دو مجموعه  $A$  و  $B$  را با هم می یابیم



Subject:

Year. Month. Date. ( )

### Composition of Operation

جایگزینی نام در یک عملیات را با هم ترکیب کنیم یعنی نام اول در ابتدا را با نام جدید که نام و آن را جواب قرار می‌دهیم را  
Select کنیم

### Rename Operation

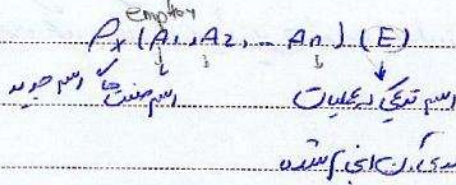
وقتی بین رابطه‌ها عملیاتی انجام می‌دهیم و جواب آن را می‌خواهیم نام آن را تغییر دهیم. یعنی entity نام جدیدی می‌دهیم  
تغییر نام رابطه را می‌توانیم تغییر دهیم

$P_x(E)$

e-id

برای تغییر نام جدول یا رابطه نام جدید را می‌توانیم تغییر دهیم

$P_x(A_1, A_2, \dots, A_n)(E)$



تاریخ: 7, 29

### Banking Example

مثالی از تبدیل اسکیم را می‌بینیم

فردان با وامی را می‌توانیم به مقدار مشخص از 1200 بیشتر کنیم

$\Pi_{loan-number} (\sigma_{amount > 1200}(loan))$

Subject:

Year. Month. Date. ( )

### Formal Definition

معنی از عملگرهای اینها استفاده می شود. یک سری عملگرهای جدید با استفاده از عملگرهای قبلی تعریف می کنیم.  
این عملگرها constant relation نام دارند.

$$\{ (L-1, B-1, 200) \text{ و } (L-2, B-2, 100) \}$$

به رابطه به صورت constant می نویسیم

constant relation ← یک سری از اینها

### Additional Operations

یک سری عملگرهای جدید نیز داریم

Set Intersection → اشتراک

Natural Join → الحاق طبیعی

### Set Intersection

اشتراک های در جدول را نشان می دهد. در اینجا باید لیسانس باشد.

$$R \cap S = r - (r - s)$$

### Natural Join

$r \bowtie s$

الحاق طبیعی

اینجا را در خروجی نشان بدهد. مقادیرهای مشترک  
آنها برابر است. (در اینجا باید لیسانس هستند)

$$r(R) \bowtie s(S) = \Pi_{R \cup S} (\sigma_{R \cap S} (r \times s))$$

آنها select می کنند مشترک

Subject:

Year. Month. Date. ( )

سوال پنجم: اسمی که هم حساب بانکی دارند و هم از بانک وام گرفته اند  
اسم آن اسمی که هم در depositor و هم در borrower بوده و هم در

$$\Pi_{c\_name} (depositor) \cap \Pi_{c\_name} (borrower)$$

اگر خاص از آن حل کنیم:

$$\Pi_{c\_name} (borrower \cap depositor)$$

اسم اسمی که هم در depositor و هم در borrower بوده و هم در

output: c-name, account-number, loan-number

به مثابه c-name با داده خروجی خاص می شود

مثال: اسمی که هم در Harrison و هم در depositor بوده و هم در borrower

$$\Pi_{branch\_name} ( \sigma_{c\_city = "Harrison"} (customer \cap account \cap depositor) )$$

c-name account-number

مثال: اسمی که هم در RAS و هم در Jain بوده و هم در depositor  
مثال: اسمی که هم در RAS و هم در Jain بوده و هم در depositor

مثال: اسمی که هم در RAS و هم در Jain بوده و هم در depositor

$$r \div s$$

نصف برش است که حاصل صفت های S در آن وجود ندارد

فرض کنید A و B دو مجموعه باشند

برای هر مقایسه ای از A، صفت مقایسه B را برش

$$R = (A_1, \dots, A_m, B_1, \dots, B_m)$$

$$S = (B_1, \dots, B_m)$$

$$r \div s = \{ t \mid t \in \Pi R \text{ and } \forall u \in S (t[u] \neq u) \}$$

Subject:

Year. Month. Date. ( )

جلسه چهارم 13.8.8

رابطه بین borrower ← customer و ban

+ خریدی است و صرفی می تواند باشد و با رابطه اول و محدودی است  
له در + صفت های له برای الیوم می رود

آنها را با هم می رانند

در رابطه هم تجربه های وجود ندارد

بر مبنای الیوم می بریم - در این صورت می توانیم برای این است

رض کنند رابطه های زیر تجربی است

stud = (S-no, S-name, S-address) ← S-no

Book = (B-no, B-name, Author) ← B-no

Book-borrow = (S-no, B-no, Date)

Course = (C-no, C-name, units)

register = (S-no, C-no, grade) ← L-no

Lecturer = (L-no, L-name)

course lecture = (L-no, C-no, room-no)

