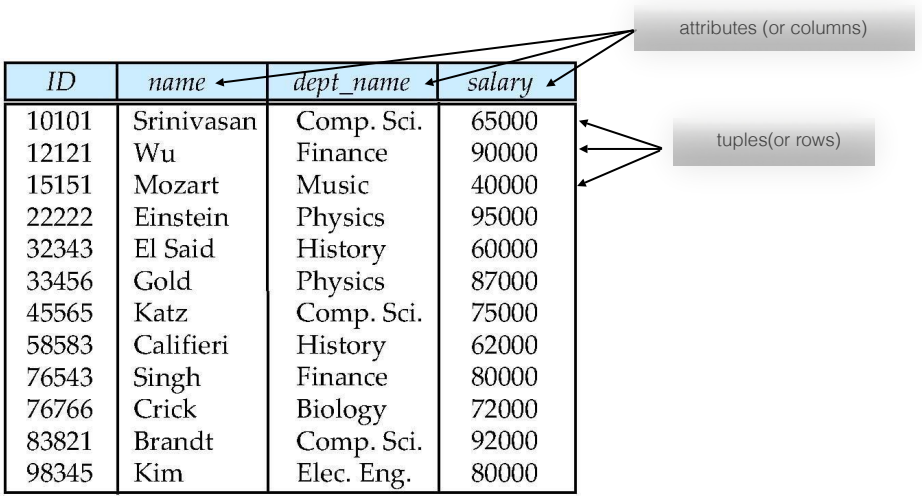
Chapter 2: Intro to Relational Model

Example of a Relation

Relation = Table

Attribute = Column

Tuple = Row



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

Attribute Types

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

(المجال - domain) يطلق على مجموعة من القيم (values) المسموح بها لكل سمة (attribute)

مثال : الsalary .. نحدد الdomain له من 4000-10000،، اذا مانقدر نضيف اقل من ال4000 ولا أكثر من ال10000

- Attribute values are (normally) required to be ***atomic***; that is, indivisible

(atomic) تعني ان تكون قيم السمة غير قابله للتجزئه (indivisible)

مثال : لو كنا نخزن الاسم الاول والاخير في الname .. هنا الصفه قابله للتجزئه بحيث يكون فيه عامود للاسم الاول فقط ،، هنا نقول انها not atomic

- The special value ***null*** is a member of every domain

كل domain متوقع ان يكون فيه قيمة فارغه (null)

- The null value causes complications in the definition of many operations

القيمة (null) تسبب تعقيدات في تعريف العديد من العمليات

Relation Schema and Instance

تمثيل الجدول بطريقة رياضيه ،، حيث أن A هي attributes وَ R هي الجدول relation schema وَ D هي الdomain

- A1, A2, …, An are attributes
- R = (A1, A2, …, An) is a relation schema //القانون//

Example:

instructor = (*ID*, *name*, *dept_name*, *salary*)

- Formally, given sets D1, D2, …. Dn a ***relation*** r is a subset of
  D1 x D2 x … x Dn
  Thus, a relation is a set of n-tuples (a1, a2, …, an) where each ai ∈ Di

الجدول (relation) عباره عن مجموعه n من الصفوف (n-tuples) المسجلة في عدد من السمات"أعمده" (a1, a2, ···, an)
حيث إن كل سمة جزء من المجال (ai ∈ Di)

- The current values (***relation instance***) of a relation are specified by a table

القيم الحالية (relation instance) من العلاقة يحدده الجدول

- An element t of r is a tuple, represented by a row in a table

Relations are Unordered //غير مرتبه//

- Order of tuples is irrelevant (tuples may be stored in an arbitrary order)
- Example: instructor relation with unordered tuples

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
76543	Singh	Finance	80000

جدول instructor.. يوضح أن الصفوف مرتبه بشكل عشوائي

Database

- A database consists of multiple relation

تتكون قاعدة البيانات من جداول متعددة ذات علاقه

- Information about an enterprise is broken up into parts (instructor - student - advisor)

يتم تقسيم المعلومات حول مؤسسة (الجامعه مثلاً) إلى أجزاء (instructor - student - advisor)

- Bad design:
  univ (instructor -ID, name, dept_name, salary, student_Id, ..) results in //مثال//
  - repetition of information (e.g., two students have the same instructor)

فيها تكرار للمعلومات لإن instructor يكون لديه أكثر من طالب .. فتتكرر معلوماته على عدد الطلاب .

  - the need for null values (e.g., represent an student with no advisor)

نحتاج للقيمة null حيث يمكن ان يكون هناك طالب لم يعين له استاذ

- Normalization theory (Chapter 7) deals with how to design "good" relational schemas

نظرية Normalization تحل مشكلة Bad design

Keys مهم**

- Let K ⊆ 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.
- Superkey K is a ***candidate key*** if K is minimal
  Example: {ID} is a candidate key for Instructor
- One of the candidate keys is selected to be the ***primary key***.
  - which one?
- ***Foreign key*** constraint: Value in one relation must appear in another
  - ***Referencing*** relation
  - ***Referenced*** relation

superkey .. هو عبارة عن column أو أكثر التي تسمح لنا بتحديد قيمه unique للجدول.

candidate key.. المفتاح المرشح .. يمكن أن يكون أي column أو مجموعة من columns التي يمكن أن تكون (unique key) في قاعدة البيانات.. يمكن أن يكون هناك عدة مفاتيح مرشحه في جدول واحد. كل مفتاح مرشح يمكن وصفها بأنه (Primary Key).

primary key.. مفتاح يميز الصف عن غيره من الصفوف .. بحيث لايمكن تكراره(unique) .. قد يكون column واحد.. أو اكثر ..

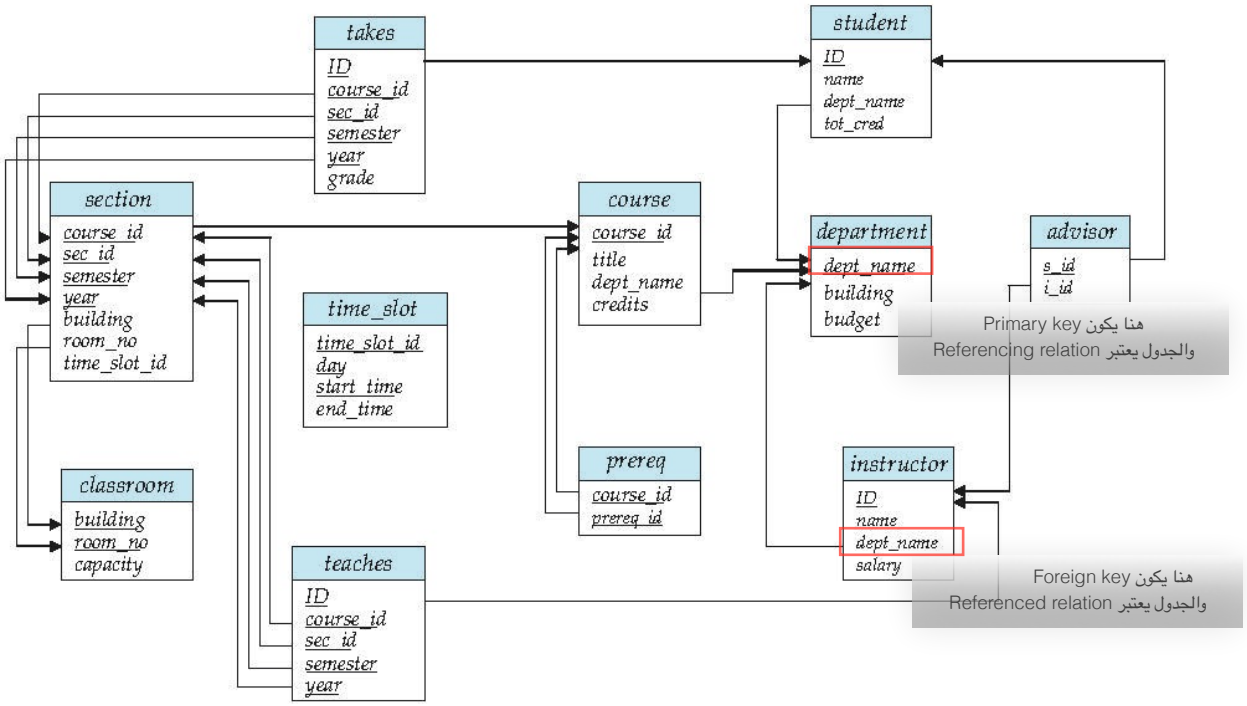
مثال على أكثر من Attribute [لوحة السياره ،، تتكون من ٣ حروف و٣ ارقام .. يسمح بتكرار الاحرف أو الارقام في سيارات اخرى .. لكن لايمكن ان يتكرر (الاحرف الثلاث+ الارقام الثلاث) في سياره اخرى .. هنا نعامل الخاصيتين معاً كأنهم خاصيه واحده تميز السياره.]

Foreign key.. هو مفتاح خارجي في جدول ما .. وهو يمثل ال(Primary Key) في جدول اخر .. بحيث يوضح العلاقة بين الجدولين.

Referencing relation .. هوالجدول اللي فيه Primary Key

Referenced relation .. هوالجدول اللي فيه Foreign key

Schema Diagram for University Database //مثال//



Relational Query Languages		
Relational operators	"Pure" languages	Procedural vs.non-procedural, or declarative
	• Relational algebra • Tuple relational calculus العمليات على الصفوف • Domain relational calculus العمليات على الأعمده	

Selection of tuples

Relation r

A	B	C	D
α	α	1	7
α	β	5	7
β	β	12	3
β	β	23	10

السؤال ... Select tuples with A=B and D > 5

السؤال رياضياً.. $\sigma_{A=B \text{ and } D > 5}(r)$

σ=Select

نحقق الشرطين في السؤال معاً.. القيم في A=B و القيم D>5

A	B	C	D
α	α	1	7
β	β	23	10

الحل

Selection of Columns (Attributes)

Relation r:

A	B	C
α	10	1
α	20	1
β	30	1
β	40	2

Select A and C -الشرط الأول

Projection (عدم التكرار)-الشرط الثاني

Π A, C (r) (رياضياً)

A	C
α	1
α	1
β	1
β	2

=

A	C
α	1
β	1
β	2

Joining two relations – Cartesian Product

Relations r

A	1
B	2
D	3
F	4
E	5

Relations s

A	1
C	2
D	3
E	4

r x s:

A	1	A	1
A	1	C	2
A	1	D	3
A	1	E	4
B	2	A	1
B	2	C	2
B	2	D	3
B	2	E	4
D	3	A	1
D	3	C	2
D	3	D	3
D	3	E	4
F	4	A	1
F	4	C	2
F	4	D	3
F	4	E	4
E	5	A	1
E	5	C	2
E	5	D	3
E	5	E	4

كل صف يتم تكراره مع كل صفوف الجدول الآخر

r = 5صفوف s = 4صفوف

r x s = 20 صف

Union of two relations الاتحاد بين جدولين

Relations r, s:

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

r ∪ s:

A	B
α	1
α	2
β	1
β	3

∪ تعني كل صفوف الجدولين معاً بدون تكرار

Set difference of two relations الطرح

Relations r, s:

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

r − s:

A	B
α	1
β	1

نحذف القيم الموجوده بالجدول r والتي تتشابه مع القيم الموجوده بالجدول s.

ننتبه لو كان (s-r) راح نتعامل مع الجدول s

Set Intersection of two relations التقاطع

A	B
α	1
α	2
β	1

r

A	B
α	2
β	3

s

s∩ r

A	B
α	2

∩ كل الصفوف المتشابهة في الجدولين

Joining two relations – Natural Join

- Let r and s be relations on schemas R and S respectively.
  Then, the "natural join" of relations R and S is a relation on schema R ∪ S obtained as follows:
  - Consider each pair of tuples tr from r and ts from s.
  - If tr and ts have the same value on each of the attributes in R ∩ S, add a tuple t to the result, where
    - t has the same value as tr on r
    - t has the same value as ts on s

Natural Join Example

A	B	C	D
α	1	α	a
β	2	γ	a
γ	4	β	b
α	1	γ	a
δ	2	β	b

r

B	D	E
1	a	α
3	a	β
1	a	γ
2	b	δ
3	b	ε

s

Natural Join
r⋈s

A	B	C	D	E
α	1	α	a	α
α	1	α	a	γ
α	1	γ	a	α
α	1	γ	a	γ
δ	2	β	b	δ

1. نبحث عن الاعمده المتشابه
2. نبحث عن القيم المتشابهه داخل هذه الاعمده فقط (كل صف من الجدول الاول مع جميع الصفوف في الجدول الثاني)
3. نجمع الصفوف
4. في الجدول الجديد ألغي الاعمده المتشابهه من الجدول الثاني
5. نكرر الخطوتين ٢و٣ مع كل صفوف الجدول الاول

الإختصارات في التشابتر

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.
⋈ (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.
× (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)
∪ (Union)	$\Pi_{name}(instructor) \cup \Pi_{name}(student)$
	Output the union of tuples from the two input relations.