IS5102
Database Management Systems
Lecture 6: Relational Model

Alexander Konovalov
alexander.konovalov@st-andrews.ac.uk

(with thanks to Susmit Sarkar)

2021

This week

- Lower level (logical) data models
- Relational data models
- Translations
- Formal query analysis

- Let A_1, A_2, . . . , A_n are attributes
- Let $R = (A_1, A_2, \ldots, A_n)$ is a relation schema
- Let $K \subseteq \{A_1, A_2, \ldots, A_n\}$
- 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?

Choosing Primary Keys

- Need to define a primary key for each table
- Sometimes a suitable set of attributes may already be present in data model
  - e.g. a consider the relation `Branch` with attributes {`branch_name, assets, branch_city`}
- Sometimes they will not ...
  - e.g. relation `Person` with {`name, age`}
  - In such cases, need to invent one or more artificial attributes which are designed to be unique
    - examples are NI number, passport number, driving licence number, NHS number, clubcard number, etc.

Foreign key constraint: Value in one relation must appear in another

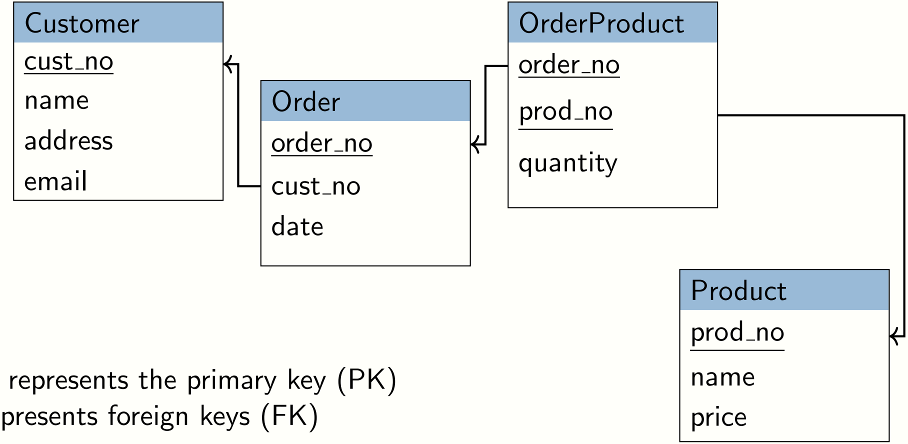
A **foreign key** constraint from attribute A of relation R_1 to the primary key B of relation R_2:

for every tuple in R_1, the value of A must also be the value of some tuple in R_2:

$$\forall v \in R_1 \quad \exists w \in R_2 \; : \; v.A = w.B$$

- A – **foreign key** from R_1 referencing R_2
- R_1 – **referencing** relation
- R_2 – **referenced** relation

Example: schema diagram



Underline represents the primary key (PK)
Arrows represents foreign keys (FK)

Exercise: how can the foreign key constraints be violated?

Reduction to Relation Schemas

- Entity sets and relationship sets can be expressed uniformly as **relation schemas** that represent the contents of the database.
- A database which conforms to an E-R diagram can be represented by a **collection of schemas**.
- For each entity set and relationship set there is a unique schema that is assigned the name of the corresponding entity set or relationship set.
- Each schema has a number of columns (generally corresponding to **attributes**), which have unique names.

Representing Entity Sets With Simple Attributes

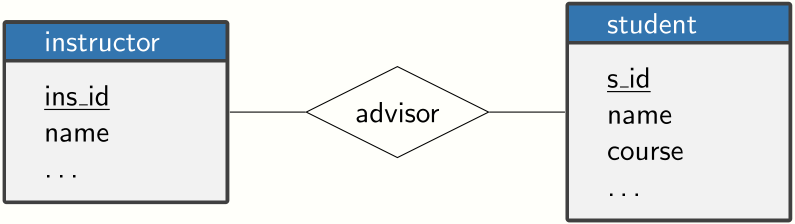
- A strong entity set reduces to a schema with the same attributes

  `student(ID, name, credits)`
- A weak entity set becomes a table that includes a column for the primary key of the identifying strong entity set

  `(building_id, floor, room_no,occupant)`

- A many-to-many relationship set is represented as a schema with attributes for the primary keys of the two participating entity sets, and any descriptive attributes of the relationship set.
- Example: schema for relationship set advisor

```
advisor = (s_id, ins_id)
```



Redundancy of Schemas

Many-to-one and one-to-many relationship sets that are **total** on the many-side:

Can be represented by adding an **extra attribute** to the "many" side, containing the primary key of the "one" side

Example: Instead of creating a schema for relationship set `inst_dept`:

add an attribute `dept_name` to the schema arising from entity set `instructor`

Composite and Multivalued Attributes

Composite attributes are **flattened out** by creating a separate attribute for each component attribute

Example: given entity set instructor with composite attribute `name` with component attributes `first_name`, `middle_initial` and `last_name`

The schema corresponding to the entity set `instructor` has the attributes `first_name`, `middle_initial` and `last_name`

Composite and Multivalued Attributes

A multivalued attribute M of an entity E is represented by a separate schema EM

- Schema EM has attributes corresponding to the primary key of E and an attribute corresponding to multivalued attribute M
- Example: Multivalued attribute `phone_number` of `instructor` is represented by a schema:

  `inst_phone = (ID, phone_number)`
- Each value of the multivalued attribute maps to a separate tuple of the relation on schema EM

- Abstract query language
- Defines a set of operations on relations
- Operations take a relation(s) as input and produce a relation as output
- They form the basis for the SQL language

Relational algebra operators include:

Selection	σ	(unary)
Projection	Π	(unary)
Cartesian Product	$\times$	(binary)
Natural Join	$\bowtie$	(binary)
Union	$\cup$	(binary)
Intersection	$\cap$	(binary)
Set difference	$-$	(binary)

Relation r:

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

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

A	B	C	D
α	α	1	7
β	β	23	10

Relation r:

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

$\Pi_{\text{A,D}}(r)$:

A	D
α	7
β	3
β	10

Set operations: union, intersection, difference

Relation r

A	B
α	1
α	2
β	1

Relation s

A	B
α	2
β	3

$r \cup s$

A	B
α	1
α	2
β	1
β	3

$r \cap s$

A	B
α	2

$r - s$

A	B
α	1
β	1

Relation r

A	B
α	1
β	2

Relation s

C	D	E
α	10	a
α	20	a
β	10	b

$r \times s$:

A	B	C	D	E
α	1	α	10	a
α	1	α	20	a
α	1	β	10	b
β	2	α	10	a
β	2	α	20	a
β	2	β	10	b

Natural Join

Relation r

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

Relation s

B	D	E
1	a	α
3	a	β
1	a	γ
2	b	δ
3	b	ϵ

$r \bowtie s$:

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

Relational Algebra Summary

Symbol	Name	Result
σ	Selection	Returns rows of the input relation that satisfy the predicate
Π	Projection	Returns the specified attributes from all rows of the input relation. Duplicate rows removed
$\times$	Cartesian product	Output all combinations of rows from the two input relations
$\bowtie$	Natural Join	Output all combinations of rows from the two input relations that are equal on their common attribute names
$\cup$	Union	Output all rows that are in the two similarly structured input relations or in both. Duplicate rows are eliminated
$\cap$	Intersection	Output all rows that are in both the two similarly structured input relations
$-$	Difference	Output all rows that are the first input relation but are not in the second

Reading and Practice

- Consolidation
  - Chapter 7, Database Design, 2nd Ed., Watt and Eng
  - Chapter 2, Database System Concepts, 6th Ed., Silberschatz, Korth and Sudarshan
  - Chapter 4 & 5.1, Database Systems, 6th Ed., Connolly, Begg

- Next few weeks: SQL
  - Chapters 15-16, Database Design, 2nd Ed., Watt and Eng
  - Chapters 3-5, Database System Concepts, 6th Ed., Silberschatz, Korth and Sudarshan
  - Chapters 6-8, Database Systems, 6th Ed., Connolly, Begg

  oin