Relational model and algebra

Feb. 24, 2023

Quick review

Relational model

Proposed in 1970 by Edgar F. Codd.

The most successful database abstraction

- Store database in simple data structures
- Access data through high-level language
- Physical storage left up to implementation

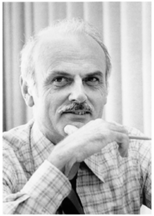

Figure: Edgar Frank Codd

https://en.wikipedia.org/wiki/Edgar_F._Codd

Relation model

- A database is a collection of relations and each relation is an unordered set of tuples (or rows).
- Each relation has a set of attributes (or columns).
- Each attribute has a name and a domain and each tuple has a value for each attribute of the relation.

ID	Artist	Year	City
1	Mozart	1756	Salzburg
2	Beethoven	1770	Bonn
3	Chopin	1810	Warsaw

Table: Artists(<u>ID</u>, Artist, Year, City)

A relational database example

ID	Album	Artist_ID	Year
1	The Marriage of Figaro	1	1786
2	Requiem Mass In D minor	1	1791
3	Für Elise	2	1867

Table: Albums(ID, Album, Artist_ID, Year)

ID	Artist	Year	City
1	Mozart	1756	Salzburg
2	Beethoven	1770	Bonn
3	Chopin	1810	Warsaw

Table: Artists(ID, Artist, Year, City)

Artist_ID	Album_ID
1	1
1	2
2	3

Table: ArtistAlbum(Artist_ID, Album_ID)

Keys

$K \subseteq \{A_1, A_2, \ldots, A_n\}$ is a superkey of schema $R(A_1, \ldots, A_n)$ if values for K are sufficient to identify a unique tuple for each possible relation instance of $R(A_1, A_2, \ldots, A_n)$.

A superkey K is a candidate key if K is minimal.

ID	Artist	Year	City
1	Mozart	1756	Salzburg
2	Beethoven	1770	Bonn
3	Chopin	1810	Warsaw

Table: Artists(<u>ID</u>, Artist, Year, City)

Primary key

A primary key is a designated candidate key of a relation.

Some DBMSs automatically create an internal primary key if we don't define one.

ID	Artist	Year	City
1	Mozart	1756	Salzburg
2	Beethoven	1770	Bonn
3	Chopin	1810	Warsaw

Table: Artists(ID, Artist, Year, City)

```
create table Artists(
        ID, vchar(8),
        Artist, vchar(20) not null,
        Year, numeric(4,0),
        City, vchar(20),
        primary key (ID)
)
```

Foreign key

A foreign key specifies that a tuple from one relation must map to a tuple in another relation.

ID	Album	Artist_ID	Year
1	The Marriage of Figaro	1	1786
2	Requiem Mass In D minor	1	1791
3	Für Elise	2	1867

Table: Albums(<u>ID</u>, Album, Artist_ID, Year)

ID	Artist	Year	City
1	Mozart	1756	Salzburg
2	Beethoven	1770	Bonn
3	Chopin	1810	Warsaw

Table: Artists(<u>ID</u>, Artist, Year, City)

Artist_ID	Album_ID
1	1
1	2
2	3

Table: ArtistAlbum(<u>Artist_ID</u>, <u>Album_ID</u>)

Foreign key (cont'd)

Foreign key constraint

The referencing attribute(s) must be the primary key of the referenced relation.

```
create table ArtistAlbum(
        Artist_ID, vchar(8),
        Albumn_ID, vchar(8),
        primary key (Artist_ID, Album_ID),
        foreign key (Artist_ID) references Artists,
        foreign key (Albumn_ID) references Albumns,
)
```

- Referencing relation: ArtistAlbum
- Referencing attributes: Artist_ID, Album_ID
- Referenced relations: Artist, Album

Relational algebra

Relational algebra

- A language for querying relational data based on fundamental relational operations.
- Each operation takes one or more relations (i.e., tables) as its input and output a new relation.
- Compose operations to make complex queries.

- Selection $\sigma_p(R)$
- Projection $\Pi_{A_1,\ldots,A_k}(R)$
- Product $R \times S$
- Union $R \cup S$
- Difference $R - S$
- Renaming $\rho_{S(A_1,\ldots,A_k)}(R)$, $\rho_S(R)$

Selection

The selection operation selects tuples that satisfy a given predicate.

- Notation:

$$\sigma_p(R)$$

- R is the input relation and p is the selection predicate.

Example

The following operation

$$\sigma_{\text{dept\_name}=\text{"Physics"}}(\text{instructor})$$

gets all the instructors in the Physics department.

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

Figure: The instructor relation

Selection (cont'd)

- Boolean connectives $=$, $\neq$, $<$, $\leqslant$, $>$ and $\geqslant$ are allowed in predicates.
- Combine predicates with logical connectives $\wedge$ (and), $\vee$ (or), $\neg$ (not).

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

Figure: The department relation

Example

- $\sigma_{\text{dept\_name}=\text{“Physics”} \wedge \text{salary}<90000}(\text{instructor})$
- $\sigma_{\text{dept\_name}=\text{building}}(\text{department})$

Projection

The projection produces from R a new relation R' that has only some of R's attributes.

- Notation:

$$\Pi_{A_1,\ldots,A_n}(R)$$

- R is the input relation and A_1, ..., A_n are attributes of R.

Example

- $\Pi_{\text{ID},\text{name},\text{salary}}(\text{instructor})$
- $\Pi_{\text{ID},\text{salary},\text{name}}(\text{instructor})$
- $\Pi_{\text{ID},\text{name},\text{salary}/12}(\text{instructor})$

ID	*name*	*salary*
10101	Srinivasan	65000
12121	Wu	90000
15151	Mozart	40000
22222	Einstein	95000
32343	El Said	60000
33456	Gold	87000
45565	Katz	75000
58583	Califieri	62000
76543	Singh	80000
76766	Crick	72000
83821	Brandt	92000
98345	Kim	80000

Projection (cont'd)

Duplicated output tuples are removed (by definition).

A	B
1	2
1	3
2	3

Table: $R(A, B)$

A
1
2

Table: $\Pi_A(R)$

B
2
3

Table: $\Pi_B(R)$

Remark. Standard relational algebra uses set semantics.

Composition of relational operations

- The input and output of an relational algebra operation are both relations.
- We can compose multiple operations into one single relational algebra expression.

Example

$$\Pi_{\text{name}}(\sigma_{\text{dept\_name}=\text{``Physics''}}(\text{instructor}))$$

Cartesian Product

The Cartesian product (or just product) of two relations R and S, denoted as

$$R \times S$$

is the set of all possible combinations of tuples from R and S.

A	B
1	2
3	4

Table: $R(A, B)$

B	C
2	6
3	8

Table: $S(B, C)$

R.A	R.B	S.B	S.C
1	2	2	6
1	2	3	8
3	4	2	6
3	4	3	8

Table: $R \times S$

Remark. For simplicity, we shall drop the relation name prefix for the attributes that appear only in R or S. E.g., we can also write $(A, R.B, S.B, C)$ as the schema of $R \times S$.

Join

Consider two tables

- instructor(ID, name, dept_name, salary)
- teaches(ID, course_id, semester, year)

We wan to find all the information about the instructors and the courses they have taught.

instructor.ID	*name*	*dept_name*	*salary*	*teaches.ID*	*course_id*	*sec_id*	*semester*	*year*
10101	Srinivasan	Comp. Sci.	65000	10101	CS-101	1	Fall	2017
10101	Srinivasan	Comp. Sci.	65000	10101	CS-315	1	Spring	2018
10101	Srinivasan	Comp. Sci.	65000	10101	CS-347	1	Fall	2017
12121	Wu	Finance	90000	12121	FIN-201	1	Spring	2018
15151	Mozart	Music	40000	15151	MU-199	1	Spring	2018
22222	Einstein	Physics	95000	22222	PHY-101	1	Fall	2017
32343	El Said	History	60000	32343	HIS-351	1	Spring	2018
45565	Katz	Comp. Sci.	75000	45565	CS-101	1	Spring	2018
45565	Katz	Comp. Sci.	75000	45565	CS-319	1	Spring	2018
76766	Crick	Biology	72000	76766	BIO-101	1	Summer	2017
76766	Crick	Biology	72000	76766	BIO-301	1	Summer	2018
83821	Brandt	Comp. Sci.	92000	83821	CS-190	1	Spring	2017
83821	Brandt	Comp. Sci.	92000	83821	CS-190	2	Spring	2017
83821	Brandt	Comp. Sci.	92000	83821	CS-319	2	Spring	2018
98345	Kim	Elec. Eng.	80000	98345	EE-181	1	Spring	2017

Figure: $\sigma_{\text{instructor.ID}=\text{teaches.ID}}(\text{instructor} \times \text{teaches})$

Theta Join

The theta join (or just join) operation allows us to combine a selection operation and a Cartesian-production operation into a single operation.

$$R \bowtie_{\theta} S = \sigma_{\theta}(R \times S)$$

Here, θ is used to referred to as the join condition.

Example

The following expressions are equivalent.

- $\sigma_{\text{instructor.ID}=\text{teaches.ID}}(\text{instructor} \times \text{teaches})$
- $\text{instructor} \bowtie_{\text{instructor.ID}=\text{teaches.ID}} \text{teaches}$

Natural Join

The natural join of R and S, written as

$$R \bowtie S$$

combines the tuples from R and S based on their common attributes.

A	B
1	2
3	4

Table: $R(A, B)$

B	C
2	6
3	8

Table: $S(B, C)$

A	B	C
1	2	6

Table: $R \bowtie S$

The schema of $R \bowtie S$ can be expressed as $(A_1, \ldots, A_i, C_1, \ldots, C_k, B_1, \ldots, B_j)$, where

- C_1, ..., C_k are the common attributes of R and S
- A_1, ..., A_i are the attributes occur in R but not S
- B_1, ..., B_j are the attributes occur in S but not R

Union

The union of R and S, denoted as

$$R \cup S$$

consists of all the tuples that appear in R or S.

Duplicated tuples are removed (by set semantics).

The union operation requires that the schema of R and S must be compatible.

- R and S must have the same arity, i.e., they have the same number of attributes.
- The attribute domains must be compatible.

Example

Find all courses taught in the Fall 2017 semester or in the Spring 2018 semester.

$$\Pi_{\text{course\_id}}(\sigma_{\text{semester}=\text{"Fall"} \wedge \text{year}=2017})(\text{section})$$
$$\cup\ \Pi_{\text{course\_id}}(\sigma_{\text{semester}=\text{"Spring"} \wedge \text{year}=2018})(\text{section})$$

course_id
CS-101
CS-315
CS-319
CS-347
FIN-201
HIS-351
MU-199
PHY-101

Difference

The difference of R and S, denoted as

$$R - S$$

consists of all the tuples appear in the table R but not in the table S.

Like the union operation, it also requires the schema of R and S to be compatible.

Example

Find the set of all courses taught Fall 2017 semester, but not in the Spring 2018 semester.

$$\Pi_{\text{course\_id}}(\sigma_{\text{semester}=\text{"Fall"} \wedge \text{year}=2017})(\text{section})$$
$$- \ \Pi_{\text{course\_id}}(\sigma_{\text{semester}=\text{"Spring"} \wedge \text{year}=2018})(\text{section})$$

course_id
CS-101

Renaming

The renaming operation

$$\rho_{S(A_1,A_2,\ldots,A_n)}(R)$$

changes the name of relation R to S. Moreover, the attributes in S are named to A_1, A_2, ..., A_n, in order from left to right.

We use $\rho_S(R)$ if we only want to rename the relation and leave the attributes intact.

A	B
1	2
3	4

Table: $R(A, B)$

B	C
2	6
3	8

Table: $S(B, C)$

X	Y	Z	W
1	2	2	6
1	2	3	8
3	4	2	6
3	4	3	8

Table: $\rho_{RS(X,Y,Z,W)}(R \times S)$

Recap

- Selection $\sigma_p(R)$
- Projection $\Pi_{A_1,\ldots,A_k}(R)$
- Product $R \times S$
- Union $R \cup S$
- Difference $R - S$
- Renaming $\rho_{S(A_1,\ldots,A_k)}(R)$, $\rho_S(R)$
- Join $R \bowtie_\theta S$, $R \bowtie S$

Exercise: let's find the highest salary

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

Figure: The instructor table

$$\Pi_{\text{salary}}(\text{instructor}) - \Pi_{\text{instructor.salary}}\left(\text{instructor} \bowtie_{\text{instructpor.salary}<\text{d.salary}} \rho_{d}(\text{instructor})\right)$$

Remark. We will NOT use such query in practice.

Extensions to relational algebra

- Relational algebra with bag semantics
- Grouping and aggregation
- ...

We will return to these when we talk about SQL in the next lecture.