ECS 165A

Database Systems

Structured Query Language (Chap. 5)

Outline

- Review Homework2- Part 1 Solutions

Practice SQL Queries

1- List all city information for cities with a population greater than 1 million.

$$\sigma_{\mathrm{pop}>1000000}(\mathrm{CITY})$$

2- What are the countryCodes of countries that have cities with a population greater than 1 million?

$$\pi_{\texttt{countryCode}}(\texttt{COUNTRY} \bowtie_{\texttt{countryCode}=\texttt{country}} \sigma_{\texttt{pop}>1000000}(\texttt{CITY}))$$

3- List all African capital cities together with their city population and the country they are located in.

$$\pi_{\texttt{cityId,CITY.pop,countryCode}}\Big(\texttt{COUNTRY} \bowtie_{\texttt{capital=cityId}} \texttt{CITY} \bowtie \sigma_{\texttt{continent='Africa'}}(\texttt{country\_continent})\Big)$$

4- What countries have a border with another country?

$$\pi_{\texttt{country\_a}}(\texttt{borders})$$

5- What countries have at least two borders with another country?

$$\pi_{\texttt{B1.country\_a}}\Big(\sigma_{\texttt{B1.country\_a=B2.country\_a} \ \wedge \ \texttt{B1.country\_b<>B2.country\_b}}\Big(\rho_{\texttt{B1}}(\texttt{borders}) \times \rho_{\texttt{B2}}(\texttt{borders})\Big)\Big)$$

6- What countries have no border with another country?

$$\pi_{\texttt{countryCode}}(\texttt{COUNTRY}) - \pi_{\texttt{country\_a}}(\texttt{borders})$$

7- List all countries in Europe and Asia.

$$\pi_{\texttt{country}}(\sigma_{\texttt{continent}=\texttt{'Europe'} \ \vee \ \texttt{continent}=\texttt{'Asia'}}(\texttt{country\_continent}))$$

Or:

$$\pi_{\texttt{country}}(\sigma_{\texttt{continent}=\texttt{'Europe'}}(\texttt{country\_continent})) \cup \pi_{\texttt{country}}(\sigma_{\texttt{continent}=\texttt{'Asia'}}(\texttt{country\_continent}))$$

8- List all countries that are located in both Europe and Asia.

$$\pi_{\texttt{country}}\left(\sigma_{\texttt{continent}=\texttt{'Europe'}}\left(\texttt{country\_continent}\right)\right)\cap \pi_{\texttt{country}}\left(\sigma_{\texttt{continent}=\texttt{'Asia'}}\left(\texttt{country\_continent}\right)\right)$$

9- What countries are located on multiple continents?

$$\pi_{\texttt{CC1.country}}(\sigma_{\texttt{CC1.country}=\texttt{CC2.country} \ \wedge \ \texttt{CC1.continent}<>\texttt{CC2.continent}} (\rho_{\texttt{CC1}}(\texttt{country\_continent}) \times \rho_{\texttt{CC2}}(\texttt{country\_continent})))$$

10- Find the city (or cities) with the largest population.

$$\pi_{\texttt{cityId}}(\texttt{CITY}) -$$
$$\pi_{\texttt{C1.cityId}}(\sigma_{\texttt{C1.pop}<\texttt{C2.pop}}(\rho_{\texttt{C1}}(\texttt{CITY}) \times \rho_{\texttt{C2}}(\texttt{CITY})))$$

11- What rivers flow in and out of the same lake?

$$\pi_{\texttt{rname}}(\texttt{river\_into\_lake} \bowtie \texttt{river\_from\_lake})$$

12- What countries have all the religions that are in the UDEF RELIGION relation?1

$$\pi_{\texttt{country}}(\texttt{country\_religion} \div \texttt{UDEF\_RELIGION})$$

1- List all countries name and the corresponding area with area not greater than 300.

```
SELECT name, area
FROM COUNTRY
WHERE area <= 300;
```

2- List all cityID and positions of capitals whose country has a population greater than 1 million.

```
SELECT cityID, lat, long
FROM CITY
WHERE cityID IN (
SELECT capital
FROM COUNTRY
WHERE pop > 1000000
);
```

3- List all capital cities together with their country names where the spoken language is 'English'.

```
SELECT cityID, COUNTRY.name
FROM CITY, COUNTRY, country_language
WHERE CITY.cityId = COUNTRY.capital
AND countryCode =
country_language.country
AND country_language.language =
'English';
```

4- List all the name of sea who is connected to another sea.

```
SELECT sea
FROM SEA
WHERE sea IN (
SELECT sea_a
FROM sea_connect
);
```

5- List all the name of sea who is connected to exactly one sea

```
(SELECT sea
FROM SEA
WHERE sea IN (
SELECT sea_a
FROM sea_connect)
)
EXCEPT (
SELECT sea
FROM SEA
WHERE sea IN (
SELECT s1.sea_a
FROM sea_connect s1, sea_connect s2
WHERE s1.sea_a = s2.sea_a
AND s1.sea_b <> s2.sea_b));
```

6- List all countries who has the religion Roman Catholic or Christian

```
SELECT countryCode
FROM COUNTRY
WHERE countryCode IN (
SELECT country
FROM country_religion
WHERE religion = 'Roman Catholic' OR
religion = 'Christian'
);
```

7- List all countries who has both religions Roman Catholic and Christian

```
SELECT countryCode
FROM COUNTRY
WHERE countryCode IN (
SELECT country
FROM country_religion
WHERE religion = 'Roman Catholic'
)
AND countryCode IN (
SELECT country
FROM country_religion
WHERE religion = 'Christian');
```

8- What countries have multiple religions

```
SELECT countryCode
FROM COUNTRY
WHERE countryCode IN (
SELECT C1.country
FROM country_religion C1,
country_religion C2
WHERE C1.country = C2.country
AND C1.religion <> C2.religion
```

Account Example

acc-id	*balance*
A1	20k
A2	18k
A3	10k

- Find the ids of the customers who have deposited into all accounts with balances larger than 15k.
- SELECT DISTINCT *cust-id*
  FROM DEPOSIT D1
  WHERE NOT EXISTS (
    (SELECT DISTINCT *acc-id*
     FROM ACC
     WHERE *balance* > 15k)
     EXCEPT
    (SELECT DISTINCT *acc-id*
     FROM DEPOSIT D2
     WHERE D1.*cust-id* = D2.*cust-id*))

DEPOSIT

acc-id	*cust-id*	*amount*
A1	1	2k
A1	1	1k
A2	1	1k
A2	2	3k
A3	3	2k
A3	2	5k

- Output answer of this query: 1
- What an inscrutable query! Lets understand it step-by-step.

- SELECT DISTINCT *cust-id*
  FROM DEPOSIT D1
  WHERE NOT EXISTS (
    (SELECT DISTINCT *acc-id*
     FROM ACC
     WHERE *balance* > 15k)
     EXCEPT
    (SELECT DISTINCT *acc-id*
     FROM DEPOSIT D2
     WHERE D1.*cust-id* = D2.*cust-id*))

acc-id	*balance*
A1	20k
A2	18k
A3	10k

DEPOSIT

acc-id	*cust-id*	*amount*
A1	1	2k
A1	1	1k
A2	1	1k
A2	2	3k
A3	3	2k
A3	2	5k

- The nested query depends on the outside query.
- So, we look at each tuple in D1, and use its *cust-id* to complete the nested query.
- Lets start with the first tuple in D1.

ACC

acc-id	*balance*
A1	20k
A2	18k
A3	10k

DEPOSIT

acc-id	*cust-id*	*amount*
A1	1	2k
A1	1	1k
A2	1	1k
A2	2	3k
A3	3	2k
A3	2	5k

- SELECT DISTINCT *cust-id* FROM DEPOSIT D1
  WHERE NOT EXISTS (
  (SELECT DISTINCT *acc-id*
  FROM ACC
  WHERE *balance* > 15k)
  EXCEPT
  (SELECT DISTINCT *acc-id*
  FROM DEPOSIT D2
  WHERE D1.*cust-id* = D2.*cust-id*))

- Lets start with the first tuple in D1. query becomes:

  (SELECT DISTINCT *acc-id*
  FROM ACC
  WHERE *balance* > 15k)
  EXCEPT
  (SELECT DISTINCT *acc-id*
  FROM DEPOSIT D2
  WHERE 1 = D2.*cust-id*))

  returns {A1, A2}

  returns {A1, A2}

- The above query returns empty. So NOT EXIST evaluates to true, and *cust-id* 1 is displayed.

ACC

acc-id	*balance*
A1	20k
A2	18k
A3	10k

DEPOSIT

acc-id	*cust-id*	*amount*
A1	1	2k
A1	1	1k
A2	1	1k
A2	2	3k
A3	3	2k
A3	2	5k

- SELECT DISTINCT *cust-id* FROM DEPOSIT D1
  WHERE NOT EXISTS (
  (SELECT DISTINCT *acc-id*
  FROM ACC
  WHERE *balance* > 15k)
  EXCEPT
  (SELECT DISTINCT *acc-id*
  FROM DEPOSIT D2
  WHERE D1.*cust-id* = D2.*cust-id*))

- Lets look at the 4th tuple in D1.

  The nested
  query becomes:

- (SELECT DISTINCT *acc-id*
  FROM ACC
  WHERE *balance* > 15k)
  EXCEPT
  (SELECT DISTINCT *acc-id*
  FROM DEPOSIT D2
  WHERE 2 = D2.*cust-id*))

returns {A1, A2}

returns {A2, A3}

- The above query returns {A1}. So NOT EXIST evaluates to false, and *cust-id* 2 is not displayed.