Structured Query Language(SQL)

Databases 2022

Postgres installation

- Option 1: Installing the environment in your local machine
  - Download and install the current version of either PostgreSQL encouraged since the reference in the labs will assume you have it. (https://www.postgresql.org/download/)
  - Start PostgreSQL server. (https://tableplus.com/blog/2018/10/how-to-start-stop-restart-postgresql-server.html)

Using docker

- Option 2: Using a docker container for postgres
  - Install the community version of Docker in your local machine.
  - Download the docker image for postgres
    - `docker pull postgres`
  - Create a docker container using the image downloaded
    - docker run -p 5432:5432 --name database2022 -e POSTGRES_PASSWORD=Innopolis$2022 -d postgres

Installing a database tool

- The database administration tool can be chosen according to your preference, but it is suggested to use Dbeaver, since it is a cross-platform tool and allows you to manage different types of databases.
- https://dbeaver.io/download/

PostgreSQL data types

- PostgreSQL supports the following data types:
  - **Boolean**
  - Character types such as **char**, **varchar**, and **text**.
  - Numeric types such as integer and floating-point number ().
  - Temporal types such as **date**, **time**, **timestamp**, and **interval**
  - **UUID** for storing Universally Unique Identifiers
  - **Array** for storing array strings, numbers, etc.
  - **JSON** stores JSON data
  - **hstore** stores key-value pair
  - Special types such as network address and geometric data.

DDL (recap)

- Create a table

```
CREATE TABLE [IF NOT EXISTS] table_name (
   column1 datatype(length) column_contraint,
   column2 datatype(length) column_contraint,
   column3 datatype(length) column_contraint,
   table_constraints
);
```

- Constraints
  - **NOT NULL,** ensures that values in a column cannot be NULL.
  - **UNIQUE**, ensure the values in a column unique across the rows within the same table.
  - **PRIMARY KEY**, a primary key column uniquely identify rows in a table. A table can have one and only one primary key.
  - **CHECK**, a check constraint ensures the data must satisfy a Boolean expression.
  - **FOREIGN KEY**, ensures values in a column or a group of columns from a table exists in a column or group of columns in another table.

Reference: https://www.postgresqltutorial.com/postgresql-create-table/

DML (Recap)

- Insert records

```
INSERT INTO table_name(column1, column2, …)
VALUES (value1, value2, …);
```

- First, specify the name of the table (table_name) that you want to insert data after the INSERT INTO keywords and a list of comma-separated columns (colum1, column2, ...).
- Second, supply a list of comma-separated values in a parentheses (value1, value2, ...) after the VALUES keyword. The columns and values in the column and value lists must be in the same order.

Exercise 1

- Consider following schema:
  - **Suppliers** (sid: integer, sname: string, address: string)
  - **Parts** (pid: integer, pname: string, color: string)
  - **Catalog** (sid: integer, pid: integer, cost: real)
- Convert the following statements in a SQL query
  - Find the names of suppliers who supply some red part.
  - Find the sids of suppliers who supply some red or green part.
  - Find the sids of suppliers who supply some red part or are at 221 Packer Street.
  - Find the sids of suppliers who supply every red or green part.
  - Find the sids of suppliers who supply every red part or supply every green part.
  - Find pairs of sids such that the supplier with the first sid charges more for some part than the supplier with the second sid.
  - Find the pids of parts supplied by at least two different suppliers.
  - find the average cost of the red parts and green parts for each of the suppliers
  - find the sids of suppliers whose most expensive part costs $50 or more

Reference: https://www.postgresqltutorial.com/postgresql-cheat-sheet/

Sample data for Exercise 1

SID	PID	Cost
1	1	$10.00
1	2	$20.00
1	3	$30.00
1	4	$40.00
1	5	$50.00
2	1	$9.00
2	3	$34.00
2	5	$48.00

Catalog

PID	Pname	Color
1	Red1	Red
2	Red2	Red
3	Green1	Green
4	Blue1	Blue
5	Red3	Red

Parts

SID	Sname	Address
1	Yosemite Sham	Devil's canyon, AZ
2	Wiley E. Coyote	RR Asylum, NV
3	Elmer Fudd	Carrot Patch, MN

Suppliers

Exercise 2

- Consider following schema:
  - Author(<u>author_id</u>, first_name, last_name)
  - AuthorPub(author_id, pub_id, author_position)
  - Book(<u>book_id</u>, book_title, month, year, editor)
  - Pub(<u>pub_id</u>, title, book_id)
- Implement the following RA in SQL queries
  - $Author \bowtie_{author_{id}=editor} Book$
  - $\Pi_{first\_name,last\_name}\left(\left(\Pi_{author\_id}(Author) - \Pi_{editor}(Book) \bowtie Author\right)\right)$
  - $\Pi_{author\_id}(Author) - \Pi_{editor}(book)$

Sample data for Exercise 2

r(author)

author_id	first_name	last_name
1	John	McCarthy
2	Dennis	Ritchie
3	Ken	Thompson
4	Claude	Shannon
5	Alan	Turing
6	Alonzo	Church
7	Perry	White
8	Moshe	Vardi
9	Roy	Batty

r(book)

book_id	book_title	month	year	editor
1	CACM	April	1960	8
2	CACM	July	1974	8
3	BTS	July	1948	2
4	MLS	November	1936	7
5	Mind	October	1950	NULL
6	AMS	Month	1941	NULL
7	AAAI	July	2012	9
8	NIPS	July	2012	9

r(pub)

pub_id	title	book_id
1	LISP	1
2	Unix	2
3	Info Theory	3
4	Turing Machines	4
5	Turing Test	5
6	Lambda Calculus	6

r(author_pub)

author_id	pub_id	author_position
1	1	1
2	2	1
3	2	2
4	3	1
5	4	1
5	5	1
6	6	1

Exercise 3

- Consider following schema:
  - Students(sid: integer, sname: string)
  - Courses(cid: integer, cname: string)
  - Registration(sid: integer, cid: integer, percent: real)
- Statement to produce RA and queries for the following statements
  - Find the distinct names of all students who score more than 90% in the course numbered 107
  - Find the number of student whose score is 75% or above in each course.
  - Find those students who are registered on no more than 2 courses.

See you next week ☺