Class 6 BigQuery

Elements of Databases

Oct 1, 2021

Announcements

- How to request more GCP credits:
  - Use a $50 coupon from both partners ($100 credits)
  - Sign up for the Free Trial ($300 credits)
  - Instapoll: how many GCP credits have you used?

Exam 1

- When: next Friday at 4pm
- Where: classroom or remote
- Duration: 60 minutes + buffer
- How: Canvas Quiz without Zoom
- Format:
  - T/F section (10-12 questions)
  - MC section (10-12 questions)
  - Coding section (3-4 questions)
- Practice Exam: see Piazza post
- Review session: Monday at 1:30pm on Zoom

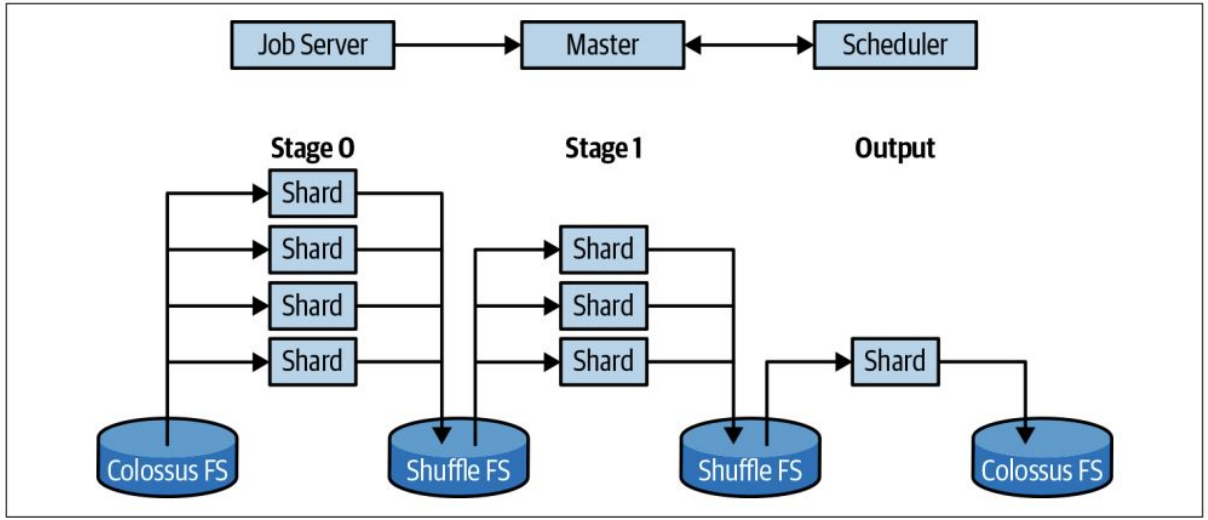
Exam Rules:

- Open-notes
- Open-book
- Open-project
- No crowdsourcing allowed
- No consulting with any humans in any form
- Piazza will be disabled

BigQuery Query Execution Pipeline

Example Query:

```
SELECT a, b, c, COUNT(*)
FROM T
GROUP BY a, b, c
ORDER BY a;
```



Source: [Google BigQuery: The Definitive Guide (2019)].

Views

- Return a table of results from a SQL query
- Saved in the database as named query
- Defined by `CREATE VIEW` statement

Employee(empid, fname, lname, job_function, level, title, manager_id, start_date,
salary, dob, ssn, emergency_contact)

```
CREATE VIEW Direct_Manager_Org AS
    SELECT empid, fname, lname, job_function, level, title,
           manager_id, start_date, salary, dob
    FROM Employee
    WHERE manager_id = 'abc'
    ORDER BY empid;
```

```
SELECT empid, fname, lname
FROM Direct_Manager_Org
WHERE start_date < '2020-01-01'
AND title = 'Data Engineer'
```

What's wrong with these queries?

Employee(empid, fname, lname, job_function, level, title, manager_id, start_date,
salary, dob, ssn, emergency_contact)

```
CREATE VIEW Director_Org AS
    SELECT empid, fname, lname, job_function, level
    FROM Employee
    WHERE level NOT IN ('SVP', 'VP', 'CEO')
    ORDER BY empid;


  SELECT empid, fname, lname
  FROM Director_Org
  WHERE salary > 300000
  AND level = 'Director';
```

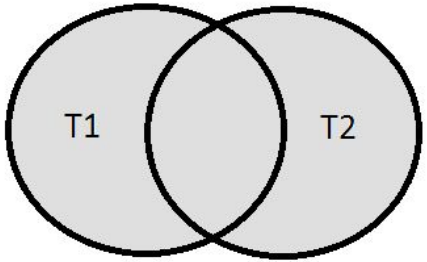
```
CREATE VIEW Senior_Manager_Org AS
    SELECT empid, fname, lname, job_function, level,
            start_date, salary
    FROM Director_Org
    WHERE level != 'Director'
    AND manager_id = 123
    ORDER BY empid;

   SELECT empid, fname, lname
   FROM Senior_Manager_Org
   WHERE start_date < '2020-01-01'
   AND job_function = 'ENG';
```

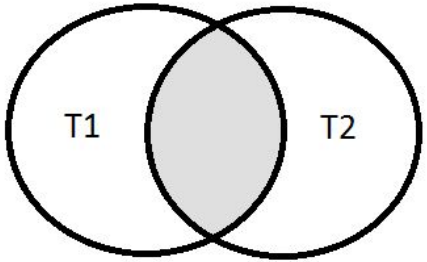
Optional Readings: Chapter 6 from Learning SQL.

Set Operations

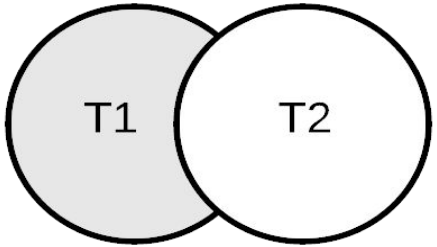
```
SELECT a, b, c FROM T1
UNION ALL | DISTINCT
SELECT a, b, c FROM T2;
```




```
SELECT a, b, c FROM T1
INTERSECT DISTINCT
SELECT a, b, c FROM T2;
```




```
SELECT a, b, c FROM T1
EXCEPT DISTINCT
SELECT a, b, c FROM T2;
```



Subqueries

```
SELECT a, b, c
FROM T1
WHERE a =
      (SELECT x FROM T2 ...)
```

Comparison Operators:

=
!=
>
<
<=
>=

- Subqueries can be attached to nearly every clause of a query
- Two major types of subqueries: uncorrelated and correlated
- Parenthesis around subquery required

Practice Question 1

Who are the oldest students?

Student(sid, fname, lname, dob, status)

Class(cno, cname, credits)

Instructor(tid, fname, lname, dept)

Takes(sid, cno, grade)

Teaches(tid, cno)

Subqueries in `WHERE` clause

```
SELECT a, b, c
FROM T1
WHERE d IN
      (SELECT x FROM T2 ...)
```

List Membership Operators:

```
IN
NOT IN
```

Practice Question 2

Who takes CS327E ***and*** CS331E?

Return sid, fname, lname.

Student(<u>sid</u>, fname, lname, dob, status)

Class(<u>cno</u>, cname, credits)

Instructor(<u>tid</u>, fname, lname, dept)

Takes(<u>sid</u>, <u>cno</u>, grade)

Teaches(<u>tid</u>, <u>cno</u>)

Practice Question 3

*Who does **not** take CS327E?*

Student(<u>sid</u>, fname, lname, dob, status)

Class(<u>cno</u>, cname, credits)

Instructor(<u>tid</u>, fname, lname, dept)

Takes(<u>sid</u>, <u>cno</u>, grade)

Teaches(<u>tid</u>, <u>cno</u>)

Subqueries in `FROM` and `JOIN` clauses

```
SELECT a, b, c
FROM (SELECT a, b, c FROM U ...)
[WHERE ...]
[ORDER BY ...]

SELECT a, b, c, d, e, f
FROM (SELECT a, b, c FROM U ...) JOIN T
ON a = d
[WHERE ... ORDER BY ...]
```

Subqueries in `HAVING` clause

```
SELECT a, b, c <aggregate functions>
FROM T1
[WHERE <boolean condition>]
GROUP BY a, b, c
HAVING <aggregate function> = (SELECT x
                               FROM T2 ...)
```

Comparison Operators: = != > < <= >=

Practice Question 4

Which classes have a higher enrollment than the overall average enrollment per class?

Return the cno and the enrollment count for those classes.

No need to account for classes with zero enrollment.

Student(sid, fname, lname, dob, status)
Class(cno, cname, credits)
Instructor(tid, fname, lname, dept)
Takes(sid, cno, grade)
Teaches(tid, cno)

Correlated Subqueries in `WHERE` clause

```
SELECT a, b, c
FROM T
WHERE c > (SELECT d FROM U WHERE U.e = T.b)
```

Comparison Operators: `=, !=, >, <, <=, >=`

List Membership Operators: `IN, NOT IN`

Subqueries in `SELECT` clause

```
SELECT a, b, c, (SELECT aggr. FROM U [WHERE U.e = T.b])
FROM T
[WHERE ... ]
```

Practice Question 4:
List all students and the highest grade received among the classes they have taken.

Student(<u>sid</u>, fname, lname, dob, status)
Class(<u>cno</u>, cname, credits)
Instructor(<u>tid</u>, fname, lname, dept)
Takes(<u>sid</u>, <u>cno</u>, grade)
Teaches(<u>tid</u>, <u>cno</u>)

Practice Question 5

Which teachers earn a higher salary than the average salary of their department?

Student(sid, fname, lname, dob, status)

Class(cno, cname, credits)

Instructor(tid, fname, lname, dept, sal)

Takes(sid, cno, grade)

Teaches(tid, cno)

Correlated Subqueries in `WHERE` clause

```
SELECT a, b, c
FROM T
WHERE EXISTS
      (SELECT * FROM U WHERE U.d = T.a)
```

Equivalent to:

```
SELECT a, b, c
FROM T JOIN U ON U.d = T.a
```

Existential Quantifiers:

```
EXISTS
NOT EXISTS
```

Practice Question 6

*Who does **not** take CS327E?*

Return the sid of all the students who do not that the class.

Student(<u>sid</u>, fname, lname, dob, status)

Class(<u>cno</u>, cname, credits)

Instructor(<u>tid</u>, fname, lname, dept)

Takes(<u>sid</u>, <u>cno</u>, grade)

Teaches(<u>tid</u>, <u>cno</u>)