

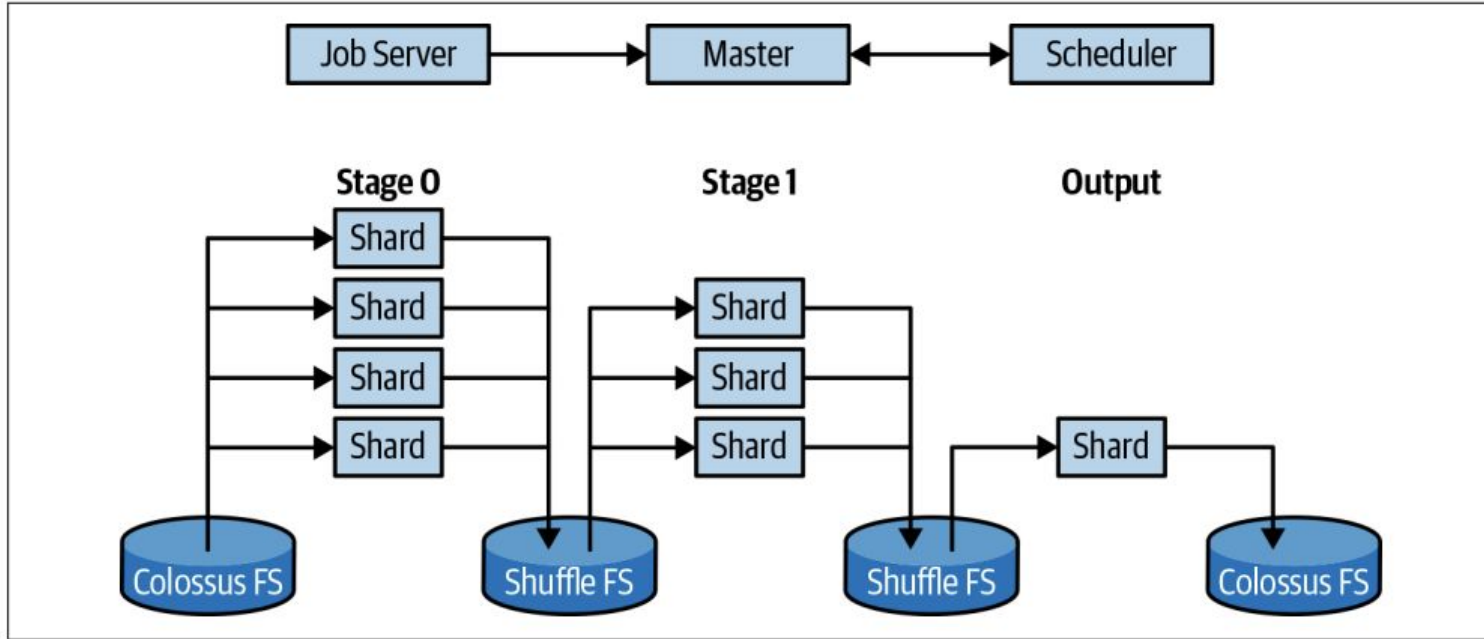
CS 327E Class 9

April 16, 2021

Announcements

- Test 3 is next class
- Review session: next Wednesday from 4pm - 5pm CT
- Milestone 2 will be due in 2 weeks

BigQuery Architecture



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

BigQuery Design Guidelines

1. A table models a single entity and an entity is modeled by a single table.
2. The collection of fields of an entity represent the attributes of that entity.
3. Each field is given a primitive type that best fits its domain of values.
4. Each table has a primary key (PK) which is made up of one or more fields that uniquely represent each record.
5. A child table has a foreign key (FK) which references its parent's PK.
6. A $m:n$ relationship is modeled as a junction table.

Common SQL Transforms

- ```
CREATE TABLE dataset.T2 AS
SELECT a, b, CAST(c as DATE) AS date FROM dataset.T1;
```
- ```
SELECT a, b, c FROM dataset.T1
UNION [DISTINCT]
SELECT x AS a, y AS b, z AS c FROM dataset.T2;
```
- ```
SELECT a, b, c, '2021-01-01' AS s FROM dataset.T1
UNION ALL
SELECT d AS b, e AS b, f AS c, '2021-02-01' AS s FROM
dataset.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 does **not** take Elements of Databases?*

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

Class(cno, cname, credits)

Instructor(tid, fname, lname, dept)

Takes(sid, cno, grade)

Teaches(tid, cno)

# 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: = != > < <= >=

# 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

# 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 3

*Who does **not** take Elements of Databases?*

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

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

Class(cno, cname, credits)

Instructor(tid, fname, lname, dept)

Takes(sid, cno, grade)

Teaches(tid, cno)

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

Class(cno, cname, credits)

Instructor(tid, fname, lname, dept)

Takes(sid, cno, grade)

Teaches(tid, cno)

## 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.*

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

Class(cno, cname, credits)

Instructor(tid, fname, lname, dept)

Takes(sid, cno, grade)

Teaches(tid, cno)



# 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)

# Database 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_View AS
 SELECT empid, fname, lname, job_function, level, title, start_date,
 salary
 FROM Employee
 WHERE manager_id = 'abc'
 ORDER BY empid;
```

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

# Example Views

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

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

```
CREATE VIEW Senior_Manager_View AS
 SELECT empid, fname, lname, job_function, level, start_date, salary
 FROM Director_View
 WHERE level != 'Director'
 ORDER BY empid;
```

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

# Data Studio Tour

- Create a BQ view
- Open [Data Studio](#)
- Create a data source (one per view)
- Create a chart and add it to a report

# Milestone 2

<http://www.cs.utexas.edu/~scohen/projects/Milestone2.pdf>