SQL: Structured Query Language
Grouping Queries

SQL SELECT statement has two more clauses to support grouping operations: `GROUP BY` and `HAVING` clauses.

GROUP BY Clause

The syntax of a `GROUP BY` clause is

```
GROUP BY <AttributeName>,...,<AttributeName>
```

The `GROUP BY` clause is added to the SELECT statement after the `WHERE` clause (or, if there is no `WHERE` clause, after the `FROM` clause.

`GROUP BY` clause causes the DBMS to separate the cartesian product of all tables referenced in the `FROM` clause into groups of tuples.

> Each group of tuples *must agree on all values* of attributes listed in the `GROUP BY` clause

Grouping operation is used to **allow for computation and reporting of aggregate operations over groups** in the SELECT statement.

Consider, for example the relational table

```
Student(FirstName, LastName, GPA, Class, Grade, School)
```

The following query

```
SELECT School, COUNT(*)
FROM Student
GROUP BY School;
```

will output the number of students enrolled in each school.

The following rules need to be observed concerning the content of the `SELECT` clause. If a `GROUP BY` clause appears in the SELECT statement, then the `SELECT` can contain only the following:

- Attributes that are listed in the `GROUP BY` clause.
- Aggregate operations on attributes not listed in the `GROUP BY` clause.
- `COUNT(*)`

HAVING Clause

`GROUP BY` clause transforms the cartesian product from a relation of tuples into a relation of groups of tuples.

`HAVING` clause to the groups is what `WHERE` clause to the individual tuples. It provides a condition, and filters out the groups that fail it.

The syntax of `HAVING` clause is:

```
HAVING <Condition>
```

Here, the condition is a boolean combination of conditions that involve:

- Attributes from the `GROUP BY` list
- Aggregate expressions over attributes **not** from the `GROUP BY` list.

For example

```
SELECT School, COUNT(*)
FROM Student
GROUP BY School
HAVING AVG(GPA) > 3.0;
```

returns the number of students enrolled in each school, which has average student GPA over 3.0.

Grouping and Aggregation in Relational Algebra

Relational Algebra includes a grouping and aggregation operation γ.

Let $R(A_1, \ldots, A_n)$ be a relation. Let L be a list $B_1, \ldots, B_k$ where B_i is either:

1. one of the attributes $A_1, \ldots A_n$ (without repetition);
2. or an expression of the form $AGG(Exp)[\longrightarrow X]$, where Exp is an expression over the set $\{A_1, \ldots, A_n\} - \{B_1, \ldots, B_k\}$ of attribute names, and AGG is one of COUNT, MIN, MAX, SUM, AVG. The optional $\longrightarrow X$ part, similarly to the renaming operator ρ establishes an alias for the aggregate expression. As a special case, $COUNT(*)$ is also allowed.

Let $L = B_1, \ldots, B_k, AGG_1(E_1) \rightarrow X_1, \ldots AGG_m(E_m) \rightarrow X_m$. The result of grouping and aggregation operation $\gamma_L(R)$ is defined below as follows:

$$\gamma_L(R) = \{t' | (\exists t \in R)((\forall i \in 1 \ldots k)(t'.B_i = t.B_i) \wedge$$
$$(\forall j \in 1 \ldots m)t'.X_j = AGG_j(t^*[E_j]) \text{ over the set } \{t^*\} \subset R, \text{ such that } \pi_{B_1,\ldots B_k}(t^*) = \pi_{B_1,\ldots,B_k}(t')\}$$

Informally, the γ operator mimics the work of the `GROUP BY` clause of the `SELECT` statement, combined with the aggregation in the `SELECT` clause. The parameter L of the grouping operator represents the list of columns in the resulting relation. There are two types of columns - attributes from table R - these form the groupings, and the aggregates of other columns (expressions constructed from other columns).

$\gamma_{B_1,\ldots,B_k,AGG_1(E_1)\rightarrow X_1,\ldots AGG_m(E_m)\rightarrow X_m}(R)$ is equivalent to

```
SELECT B1, ..., Bk, AGG1(E1) AS X1, ..., AGGm(Em) AS Xm
FROM R
GROUP BY B1, ..., Bk;
```

To remove some of the $B_1, \ldots, B_k$ from the result, use projection.

The `HAVING` clause of the SQL `SELECT` statement is modeled via the selection operation applied to the result of the grouping an aggregation operation. In particular,

```
SELECT B1, ..., Bk, AGG1(E1) AS X1, ..., AGGm(Em) AS Xm
FROM R
GROUP BY B1, ..., Bk
HAVING C;
```

is equivalent to $\sigma_C(\gamma_{B_1,\ldots,B_k,AGG_1(E_1)\rightarrow X_1,\ldots AGG_m(E_m)\rightarrow X_m}(R))$.