**lec 2 CMU**

-**Relational Languages**

**important:**SQL is based on **bags** (duplicates) not sets(no duplicates ) .

-**sql history**:-SQL is not a dead language. It is being updated with new features every couple of years.

**-Aggregations:-Functions that return a single value from a bag of**

**tuples:**

**→ AVG(col)→ Return the average col value.**

**→ MIN(col)→ Return minimum col value**

**→ MAX(col)→ Return maximum col value.**

**→ SUM(col)→ Return sum of values in col.**

**→ COUNT(col)→ Return # of values for col**

**\***COUNT, SUM, AVG support DISTINCT

**+group by:-**Project tuples into subsets and calculate aggregates against each subset.

-Having:-Filters results based on aggregation computation. Like a WHERE clause for a GROUP BY

**-string operations:-LIKE** is used for string matching.

String-matching operators

→'%' Matches any substring (including

empty strings).

→'\_' Match any one character

**-Date Time Operations:-**Operations to manipulate and modify DATE/TIME attributes.

Can be used in both output and predicates.

Support/syntax varies wildly…

Demo: Get the # of days since the beginning of

the year.

**-Output control :-**

**-**ORDER BY <column\*> [ASC|DESC]

→ Order the output tuples by the values in one or more of

their co

-LIMIT <count> [offset]

→ Limit the # of tuples returned in output.

→ Can set an offset to return a “range”.

**-Redirection:-**Store query results in another table:

→ Table must not already be defined.

→ Table will have the same # of columns with the same

types as the input.

Insert tuples from query into another table:

→ Inner SELECT must generate the same columns as the

target table.

→ DBMSs have different options/syntax on what to do with

integrity violations (e.g., invalid duplicates).

-NESTED QUARIES:-Queries containing other queries.

They are often difficult to optimize.

Inner queries can appear (almost) anywhere in query.

**ALL**→ Must satisfy expression for all rows in the

sub-query.

**ANY**→ Must satisfy expression for at least one row

in the sub-query.

**IN**→ Equivalent to '=ANY()' .

**EXISTS**→ At least one row is returned.ery.

**-Window Functions:-**Performs a "sliding" calculation across a set of

tuples that are related.

Like an aggregation but tuples are not grouped

into a single output tuples**.**

**Aggregation functions:**

→ Anything that we discussed earlier

Special window functions:

→ ROW\_NUMBER()→ # of the current row

→ RANK()→ Order position of the current

row.

The OVER keyword specifies how to

group together tuples when

computing the window function.

Use PARTITION BY to specify group.

-**Common Table Expressions:-**

Provides a way to write auxiliary statements for

use in a larger query.

→ Think of it like a temp table just for one query.

Alternative to nested queries and views.

-You can bind output columns to names before the

AS keyword.

**Note**:-You should (almost) always strive to compute your

answer as a single SQL statement.