MEETING 2

- Query Section

1. Partitioning

| usage | : to create partition of some database table over certain rules |
| --- | --- |
| query | -- create the initial table  CREATE TABLE [table\_name] (  column\_name column\_type,  column\_name column\_type,  column\_name column\_type,  column\_name column\_type  ) PARTITION BY [ RANGE | LIST | HASH ]  -- create partition manually  CREATE TABLE [partition\_name] PARTITION OF [table\_name]  FOR VALUE IN ([target\_value]) |

1. Pruning Partition

| usage | : to enable queries to filter partitions while running queries |
| --- | --- |
| query | SET enable\_partition\_pruning = on; |

1. Common Table Expression

| usage | : create temporary table to help further complex process |
| --- | --- |
| query | -- CTE  WITH [cte\_table\_name]  AS (  SELECT  certain\_column,  certain\_column,  certain\_column  FROM [initial\_table]  )  SELECT \*  FROM [cte\_table\_name]  WITH [condition\_clause] |

1. Creating Function

| usage | : user defined function |
| --- | --- |
| query | – Function  CREATE OR REPLACE FUNCTION function\_name() RETURNS trigger AS $$  BEGIN  [what function do];  RETURN NULL;  END;  $$ LANGUAGE plpgsql; |

1. Materialized View

| usage | : store object of query result that easily access and updated |
| --- | --- |
| query | -- MV  CREATE MATERIALIZED VIEW [view\_table\_name]  AS  SELECT some\_column, some\_aggregation, some\_operation  FROM [source\_table]  GROUP BY [column]  ORDER BY [column] |