Databases tried out

1. PostgresSQL
2. Oracle
3. SparkSQL
4. And read about others, but didn't try much out. Very few seemed to support automatic query rewrite

Materialized views in Hive

All features for materialized views i.e.

- Automatic Query Rewrite
- Support Materialized View
- Explain and Analyze

https://cwiki.apache.org/confluence/display/Hive/Materialized+views

Implementation details

- Tried databases to check for automatic query rewrite capabilities
- Narrowed in on Apache Hive with Apache Calcite adapter for automatic query rewrite
- Setup hadoop and hive
- Downloaded and looked into IMDB dataset
- Tried to load it and failed. Will try to do soon.

Paper details

1.Opportunistic View Materialization with Deep Reinforcement Learning

-

2.DynaMAT

- Deals with dynamic view creation
- Selection to minimise latency
- Mainly for usage patterns that are constantly evolving

3.RECYCLER

- Deals with materializing all the intermediate results
- Shows how recycling can be applied in pipelined query executors.
- Focus is on choosing the ones making best use of limited intermediate cache
- Also caching of results during query evaluation for possible reuse in future workloads as recycling

4.WATCHMAN (War

- It is a design of an intelligent cache manager for sets retrieved by queries; which is particularly well suited for data warehousing environment
- It makes use of mainly two algorithms : Cache Replacement Algorithm and Cache Admission algorithm
- The algorithms mainly aim at minimizing the execution time of queries that miss the cache instead of minimizing the hit ratio, as is the case in buffer management.
- It provides hints to the buffer manager by instructing it to evict those pages which are used mostly by queries whose retrieved sets are cached. Such hints, if correct, may free the buffer space faster and thus improve the buffer manager's performance.

Environment Design

Environment should contain two main components wrt implementation, **state representation** and ***state action representation***. We would like to follow the same representation as followed in the paper, mainly due to lack of another approach, and because we know this approach has worked in the past.

State representation -

State representation depends on the view representation

View representation -
[-------------- TABLES AS features ----------------]
[1 0 0 0 0 1 1 1 0 0]

Where a 1 describes that the respective table is participating in a join within the represented view, and 0 represents otherwise

State representation -
State is a collection of various view in the system. This can be represented as

S = MV1 OR MV2 OR …. OR MVn

Action -
Action is basically a single view, that should or should not be created. Could be anything.

Coding stuff (Commands for setting up Hadoop and Hive, important concepts, things to remember)

- **Apache Hive**
- **Hadoop**

QUESTIONS ?

(PLEASE ADD IN ANY. NO QUESTION IS TOO STUPID)

-