

Goal 1

Achieve Stable Query Execution Performance

- Upgrades make 99% of the queries happy
- Yet 1% might be sad
- Business critical queries need stable performance

Robustness & Flexibility

Goal 2

Tell the optimizer you have better plans

- Query optimization is never perfect
- Human in the loop can help

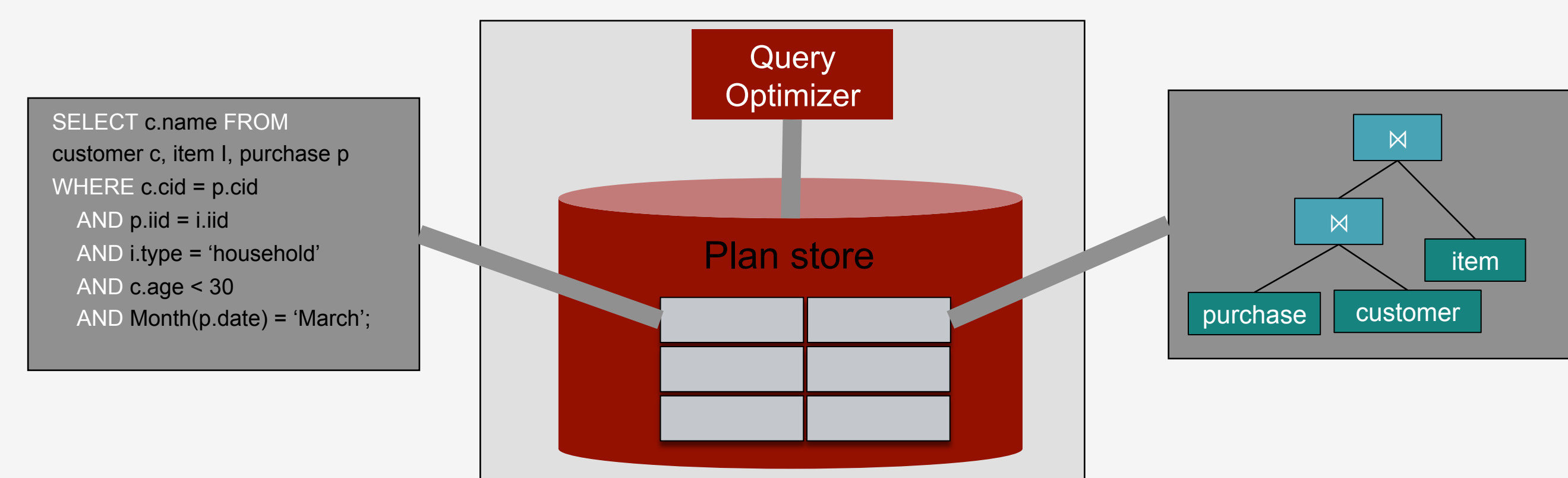


Storing Execution Plans

Each execution plan is stored as a SQL Query annotated with hints

| Original Query | Execution Plan | Annotated Query |
|--|----------------|--|
| <pre>SELECT c.name FROM customer c, item i, purchase p WHERE c.cid = p.cid AND p.iid = i.iid AND i.type = 'household' AND c.age < 30 AND Month(p.date) = 'March';</pre> | | <pre>SELECT /*+ syntactic_join */ c.name FROM ((purchase p JOIN /*+ Jtype(H) */ customer c ON c.cid = p.cid) JOIN /*+ Jtype(H) */ item i ON p.iid = i.iid) WHERE i.type = 'household' AND c.age < 30 AND Month(p.date) = 'March';</pre> |

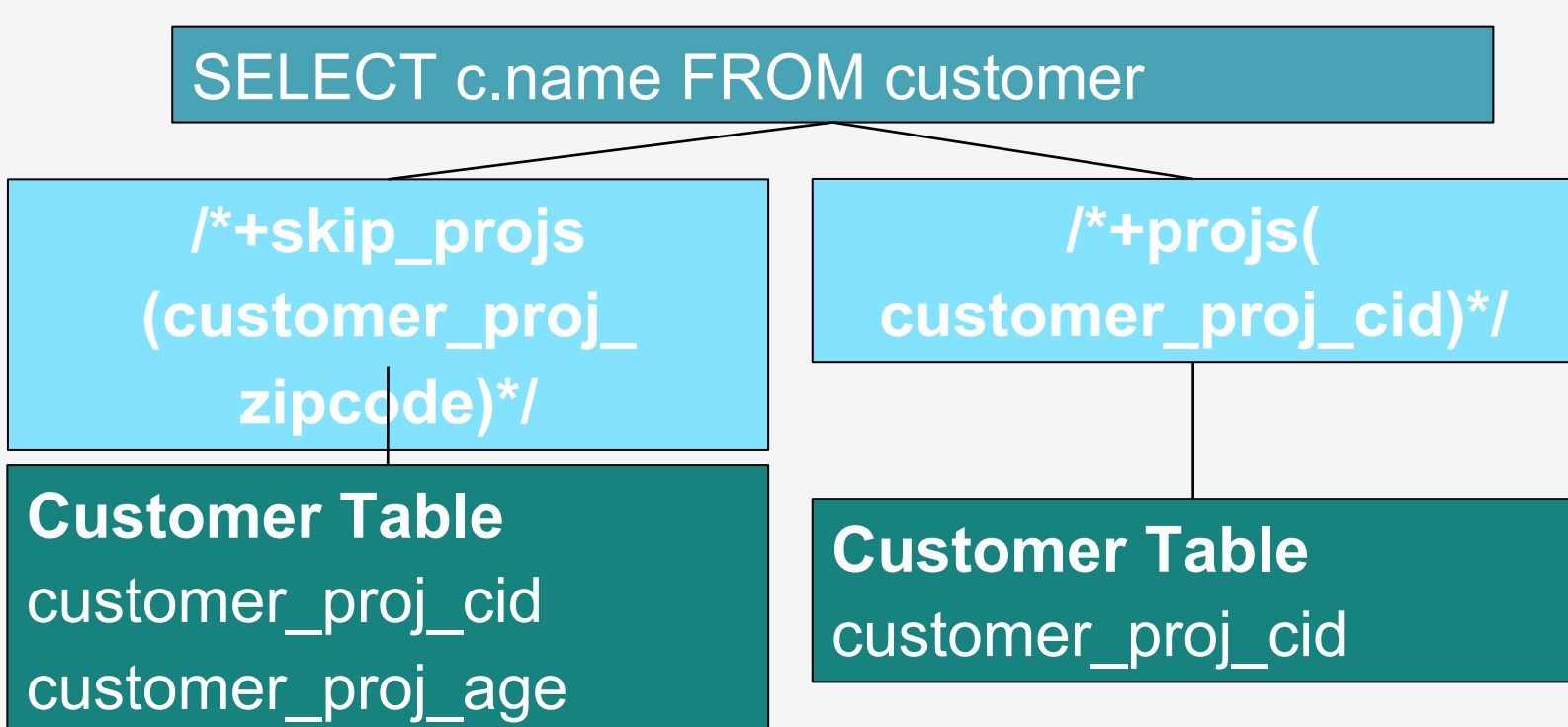
Plan store of directed queries



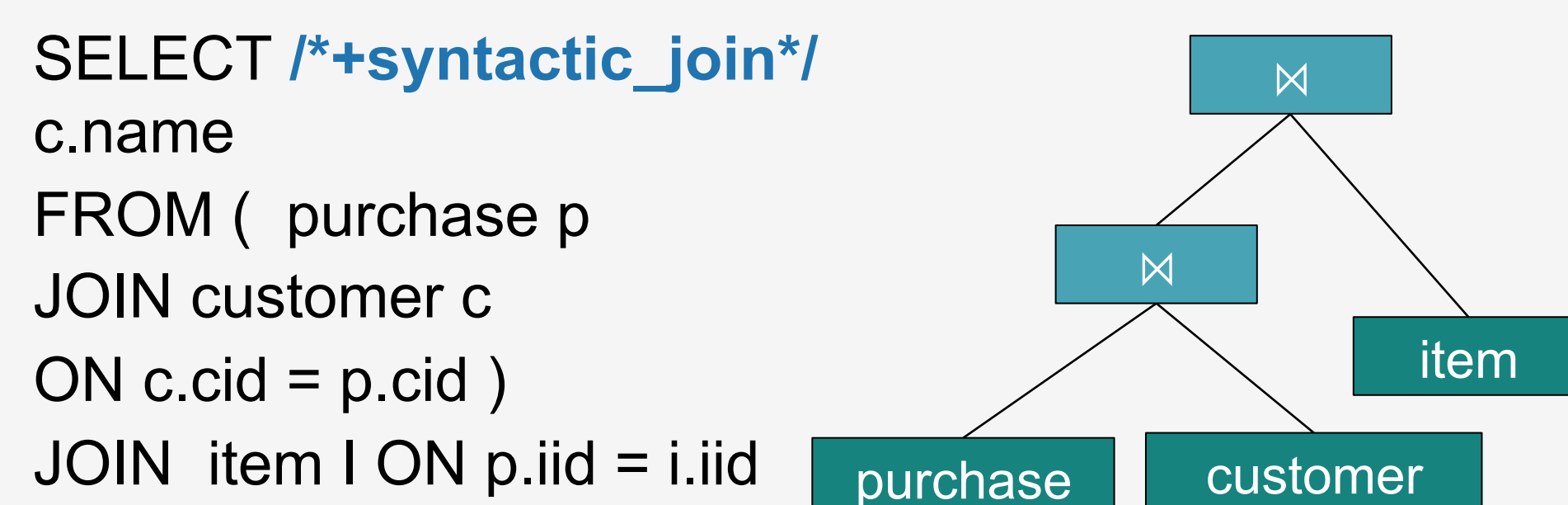
Plan look-up happens when the directed query is activated

Annotated SQL and HINTS

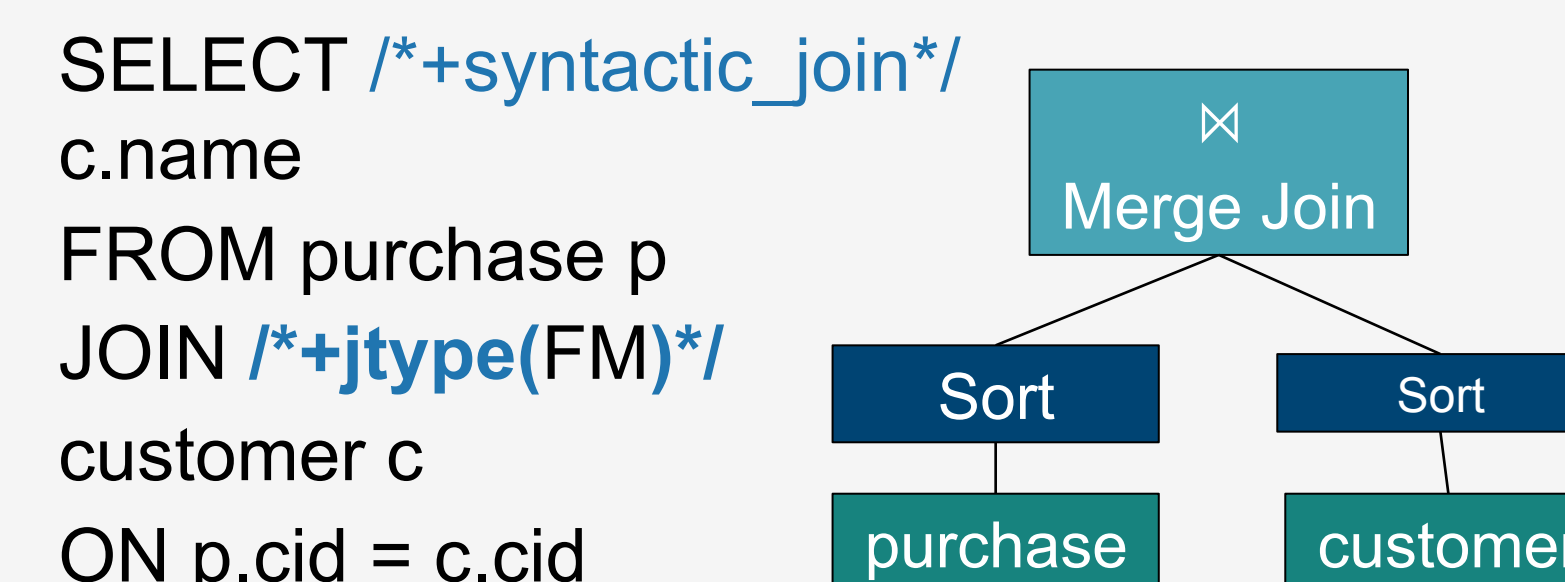
Hints to force projection choice



Hint to force inner/outer and join order



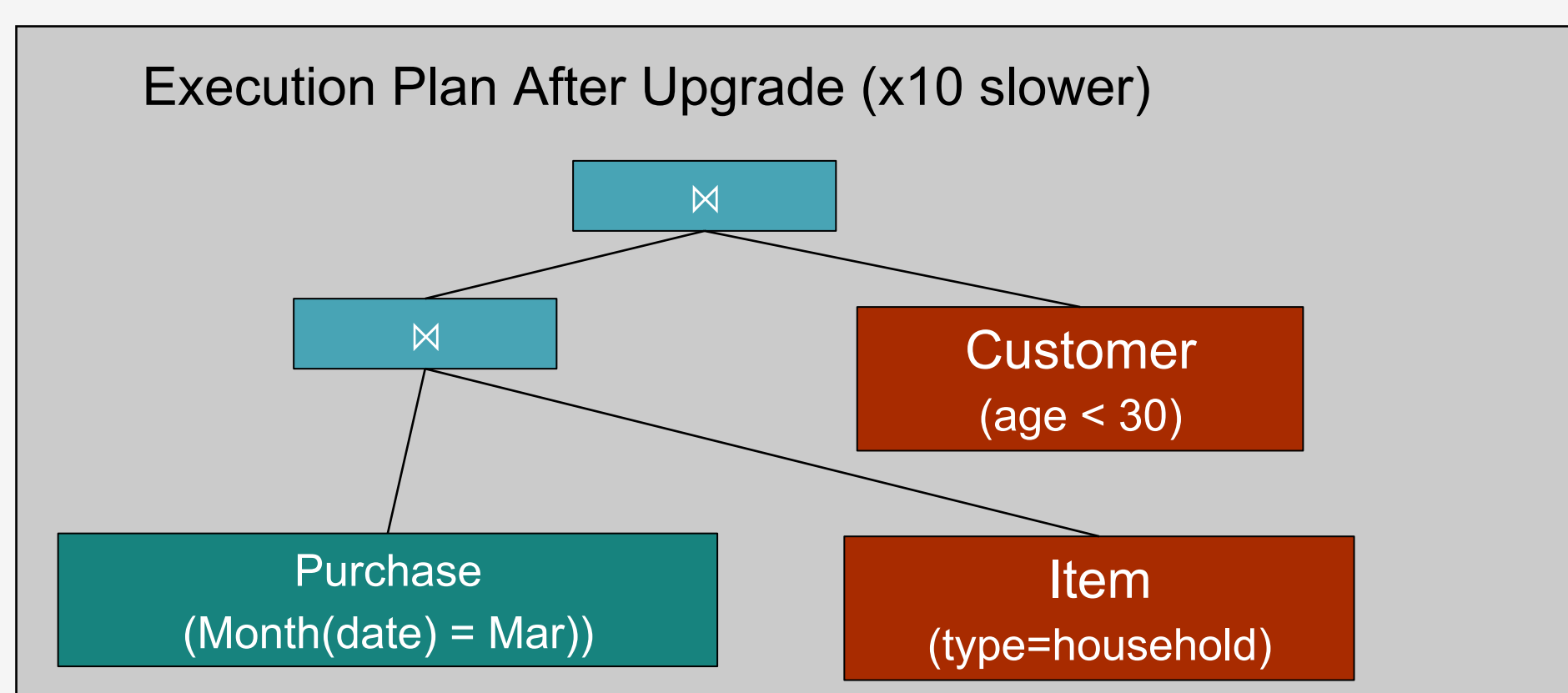
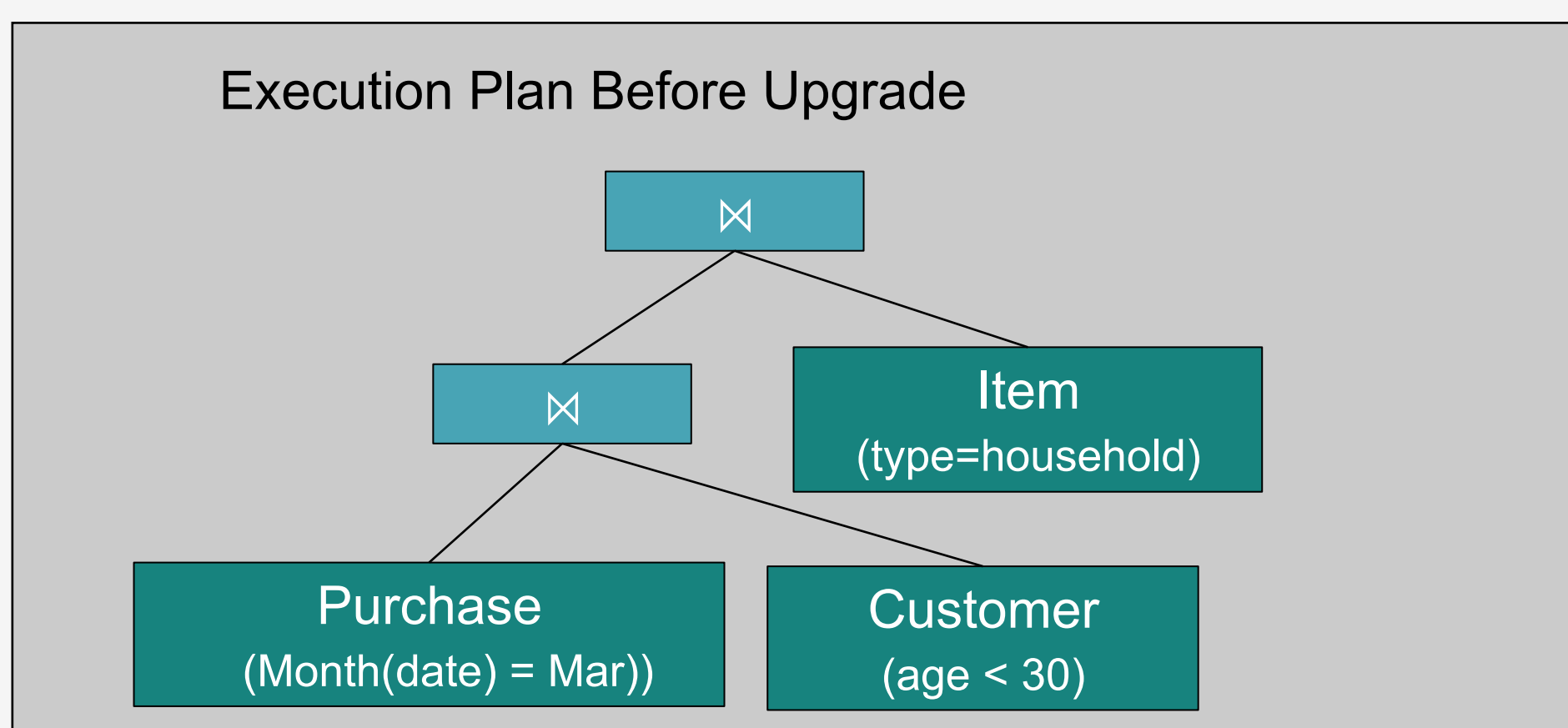
Hint to force join algorithm



Hint to force data distribution

- L (local)
- B (broadcast)
- R (resegment on join key)
- F (filter)

Stabilizing Query Plans



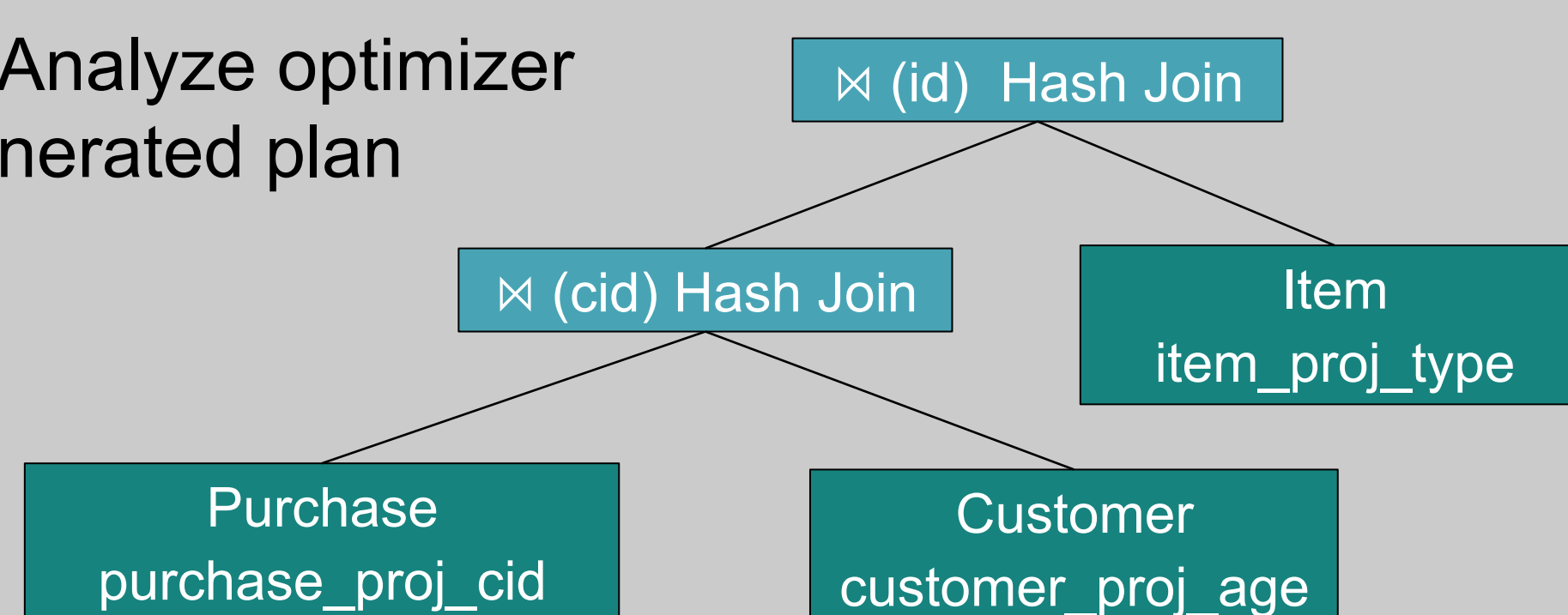
Before upgrade:

```
CREATE DIRECTED QUERY OPTIMIZER optPlan
SELECT c.name FROM customer c, item i, purchase p
WHERE c.cid = p.cid AND p.iid = i.iid AND i.type = 'household'
AND c.age < 30 AND Month(p.date) = 'March';
```

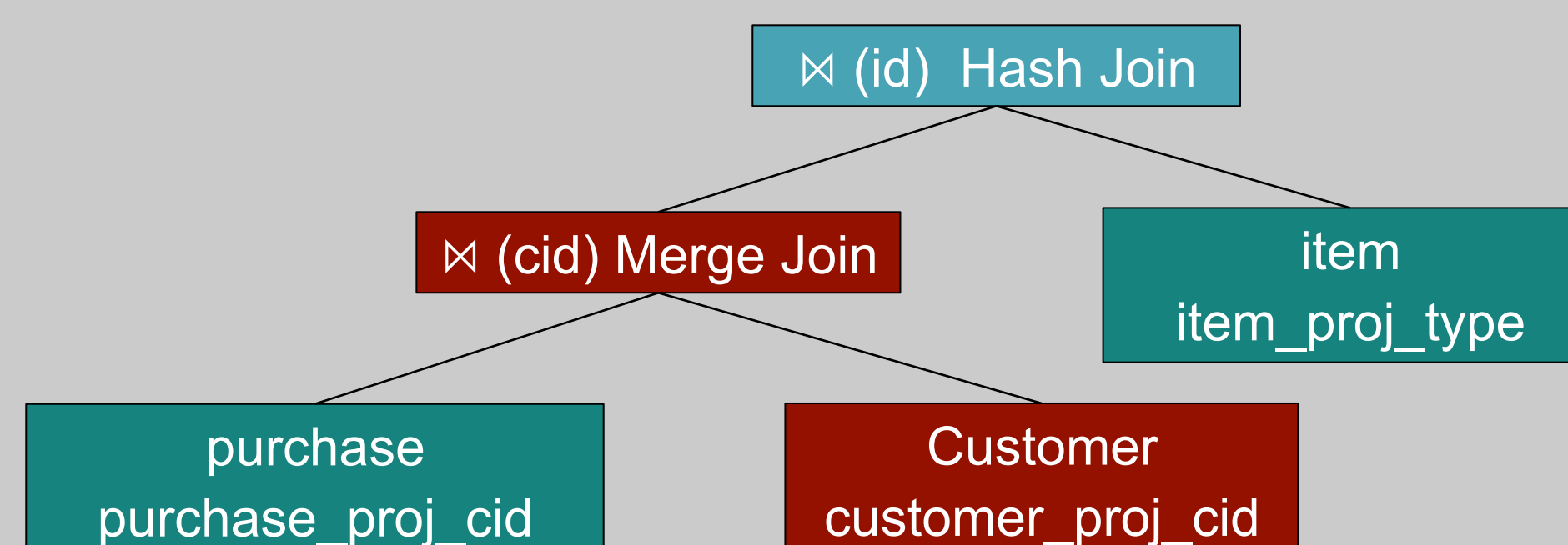
After upgrade: **ACTIVATE DIRECTED QUERY** optPlan

Tell the optimizer you have better plans

1. Analyze optimizer generated plan



3. Use hints to create a better plan



2. Generate annotated SQL for current plan

```
SELECT /*+ syntactic_join */ c.name
FROM (( purchase p /*+projs('purchase_proj_cid')*/
JOIN /*+ JType(H) */ customer c /*+projs('customer_proj_age')*/
ON c.cid = p.cid)
JOIN /*+ JType(H) */ item i /*+projs('item_proj_type')*/ ON p.iid = i.iid)
WHERE i.type = 'household' AND c.age < 30 AND Month(p.date) = 'March';
```

4. Create Customized Execution Plans

```
CREATE DIRECTED QUERY CUSTOM customPlan
SELECT /*+ syntactic_join */ c.name
FROM ((purchase p /*+projs('purchase_proj_cid')*/
JOIN /*+ JType(M) */ customer c /*+projs('customer_proj_cid')*/
ON c.cid = p.cid)
JOIN /*+ JType(H) */ item i /*+projs('item_proj_type')*/ ON p.iid = i.iid)
WHERE i.type = 'household' AND c.age < 30 AND Month(p.date) = 'March';
```

Flexible Query Execution Plans

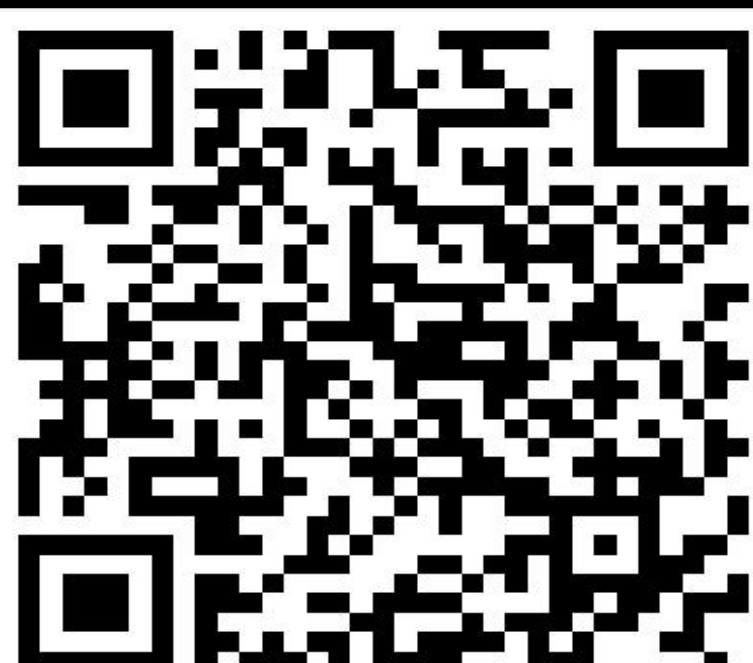
```
CREATE DIRECTED QUERY OPTIMIZER
SELECT ... FROM Purchase, ...
WHERE Month(p.date) = 'March' /*+IgnoreConst(1)*/;
```

```
SELECT ... FROM Purchase, ...
WHERE Month(p.date) = 'March'
```

```
Purchase
(Month(date) = March))
```

```
SELECT ... FROM Purchase, ...
WHERE Month(p.date) = 'April'
```

```
Purchase
(Month(date) = April))
```



Want to join the
Vertica Query Engine team?

Benefits Achieved

- ✓ Achieve stable query execution performance for business-critical queries between Vertica upgrades
- ✓ Improve query performance using optimizer hints
- ✓ Achieve flexibility for both Vertica users and developers