SQL – Subqueries

Subquery = Nested Query = Inner Query

- A complete SQL query that exists within a larger SQL query

- Larger query is the Outer Query or Main Query

Why Subqueries?

- May want to isolate steps

- They are necessary to complete some tasks

- May have better performance

SQL – Subqueries

Common Use #1: WHERE clause using IN operator

Without Subquery – list of static values:

```
SELECT *
FROM TABLE_A
WHERE FIELD_A IN ('VALUE_1','VALUE_2',...,'VALUE_N')
```
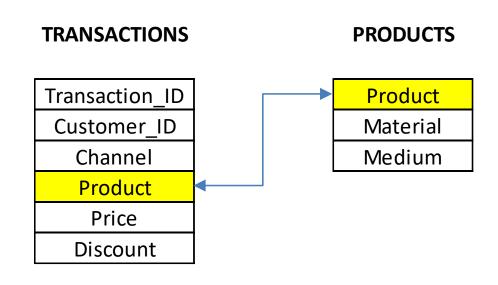
With Subquery – list of values from query on TABLE_B:

```
SELECT *
FROM TABLE_A
WHERE FIELD_A IN (SELECT FIELD_B FROM TABLE_B)
```

SQL – Subqueries

Transaction_ID	Customer_Id	Channel	Product	Price	Discount
1000123	60067	Web	Book	9.95	
1000124	12345	Store	Book	11.95	
1000125	23451	Store	DVD	14.95	
1000126	70436	Reseller	DVD	19.95	5
1000127	66772	Store	Magazine	3.25	
1000128	60067	Web	Book	29.95	
1000129	72045	Web	DVD	9.95	
1000130	82371	Reseller	Magazine	2.5	0.25
1000131	12345	Store	Book	7.95	

Product	Material	Medium
Book	Stock Paper	Visual
DVD	Plastic	Audiovisual
Magazine	Glossy Paper	Visual
CD	Plastic	Audio
Newspaper	Newsprint	Visual
MP3	Digital	Audio

TRANSACTIONS

PRODUCTS

TRANSACTIONS
Transaction_ID
Customer_ID
Channel
Product
Price
Discount

PRODUCTS
Product
Material
Medium

Be Boulder.

University of Colorado **Boulder**

SQL – JOIN Statements

Let's say we wanted l list of purchases of Medium 'Visual' only

Without Subquery – join and filtering approach:

```sql
SELECT a.*
FROM TRANSACTIONS a
LEFT JOIN PRODUCTS b
ON a.PRODUCT = b.PRODUCT
WHERE b.MEDIUM = 'Visual'
```

With Subquery – list of values from query on PRODUCTS:

```sql
SELECT *
FROM TRANSACTIONS
WHERE PRODUCT IN
   (SELECT PRODUCT FROM PRODUCTS WHERE MEDIUM = 'Visual')
```

Be Boulder.

University of Colorado **Boulder**

SQL – JOIN Statements

Both queries yield the same results:

Transaction_ID	Customer_Id	Channel	Product	Price	Discount
1000123	60067	Web	Book	9.95	
1000124	12345	Store	Book	11.95	
1000127	66772	Store	Magazine	3.25	
1000128	60067	Web	Book	29.95	
1000130	82371	Reseller	Magazine	2.5	0.25
1000131	12345	Store	Book	7.95	

SQL – Subqueries

Common Use #2: Replace a table reference in a query, especially when a JOIN command is used

Without Subquery – basic join query:

```
SELECT a.FIELD_1, ..., a.FIELD_N, b.FIELD_1, ..., b.FIELD_N
FROM TABLE_1 a
LEFT JOIN TABLE_2 b
ON a.KEY = b.KEY
```

With Subquery – reference to table 2 replaced by a Subquery:

```
SELECT a.FIELD_1, ..., a.FIELD_N, b.FIELD_1, ..., b.FIELD_N
FROM TABLE_1 a
LEFT JOIN
 (SELECT KEY, SUM(FIELD_A) as FIELD_1, ...., COUNT(*) as FIELD_N
  FROM TABLE_BGROUP BY KEY) b
ON a.KEY = b.KEY
```

SQL – JOIN Statements

Let's say we wanted to add total sales by product to our PRODUCT table

```
SELECT a.*, b.TOTAL_SALES
FROM PRODUCTS a
LEFT JOIN (SELECT PRODUCT, SUM(PRICE) as TOTAL_SALES
           FROM TRANSACTIONS
           GROUP BY PRODUCT) b
ON a.PRODUCT = b.PRODUCT
```

Product	Material	Medium	TOTAL_SALES
Book	Stock Paper	Visual	59.80
DVD	Plastic	Audiovisual	44.85
Magazine	Glossy Paper	Visual	5.75
CD	Plastic	Audio	0.00
Newspaper	Newsprint	Visual	0.00
MP3	Digital	Audio	0.00