

Murach Chapter 5 Part 1

How to Code Summary Queries

Week 5

Knowledge Points in this lecture

- Basic Summary Queries
 - AVG, SUM, COUNT, MAX, MIN
- DISTINCT in summary queries

Aggregation Function

- Scalar Functions
 - Operate on a single value
 - E.g. SYSDATE, ROUND, TO_CHAR, MOD
 - Also called **single row** function
- Aggregate functions
 - Compute a summary value for multiple values in a set of selected rows
 - E.g. AVG, SUM, COUNT, MAX, MIN
 - Also called **column** function
- Summary query
 - SELECT using aggregate functions

The syntax of the aggregate functions

AVG ([ALL | DISTINCT] expression)
SUM ([ALL | DISTINCT] expression)
MIN ([ALL | DISTINCT] expression)
MAX ([ALL | DISTINCT] expression)
COUNT ([ALL | DISTINCT] expression)
COUNT (*)

- [A]: A is optional
- A | B: A or B
- A: A is default

- ALL:
 - Default option; include unique and duplicate values (**NULL excluded**)
- DISTINCT
 - Include only unique values in computation (**NULL excluded**)
 - Typically used only in COUNT(...)
- * - everything including: NULL, duplicate and unique values
- COUNT(*) – number of rows in the query result

Extra Summary Queries

- Extra Summary Queries using table testagg
 - In file: MurachCh5ExtraSQL-SummaryQuery-Basics-Recorded.sql
 - Need to create table testagg using the sql script file MurachCh5-CreateTestAGG.sql.

A summary query

```
SELECT COUNT(*) AS number_of_invoices,  
       SUM(invoice_total - payment_total - credit_total)  
       AS total_due  
FROM invoices  
WHERE invoice_total - payment_total - credit_total > 0
```

The result set

| | NUMBER_OF_INVOICES | TOTAL_DUE |
|---|--------------------|-----------|
| 1 | 40 | 66796.24 |

A summary query with COUNT(*), AVG, and SUM

```
SELECT 'After 1/1/2008' AS selection_date,  
       COUNT(*) AS number_of_invoices,  
       ROUND(AVG(invoice_total), 2) AS avg_invoice_amt,  
       SUM(invoice_total) AS total_invoice_amt  
FROM invoices  
WHERE invoice_date > '01-JAN-2014'
```

The result set

| | SELECTION_DATE | NUMBER_OF_INVOICES | AVG_INVOICE_AMT | TOTAL_INVOICE_AMT | |
|---|----------------|--------------------|-----------------|-------------------|--|
| 1 | After 1/1/2014 | 114 | 1879.74 | 214290.51 | |

A summary query with MIN and MAX functions

```
SELECT 'After 1/1/2014' AS selection_date,  
       COUNT(*) AS number_of_invoices,  
       MAX(invoice_total) AS highest_invoice_total,  
       MIN(invoice_total) AS lowest_invoice_total  
FROM invoices  
WHERE invoice_date > '01-JAN-2014'
```

The result set

| | SELECTION_DATE | NUMBER_OF_INVOICES | HIGHEST_INVOICE_TOTAL | LOWEST_INVOICE_TOTAL | |
|---|----------------|--------------------|-----------------------|----------------------|--|
| 1 | After 1/1/2014 | 114 | 37966.19 | 6 | |

A summary query for non-numeric columns

```
SELECT MIN(vendor_name) AS first_vendor,  
       MAX(vendor_name) AS last_vendor,  
       COUNT(vendor_name) AS number_of_vendors  
FROM vendors
```

The result set

| | FIRST_VENDOR | LAST_VENDOR | NUMBER_OF_VENDORS |
|---|--------------|--------------|-------------------|
| 1 | ASC Signs | Zylka Design | 122 |

A summary query with the **DISTINCT** keyword

```
SELECT COUNT(DISTINCT vendor_id) AS number_of_vendors,  
       COUNT(vendor_id) AS number_of_invoices,  
       ROUND(AVG(invoice_total),2) AS avg_invoice_amt,  
       SUM(invoice_total) AS total_invoice_amt  
FROM invoices  
WHERE invoice_date > '01-JAN-2014'
```

The result set

| | NUMBER_OF_VENDORS | NUMBER_OF_INVOICES | AVG_INVOICE_AMT | TOTAL_INVOICE_AMT | |
|---|-------------------|--------------------|-----------------|-------------------|--|
| 1 | 34 | 114 | 1879.74 | 214290.51 | |