SQL: Advanced Queries

- Chapter 6 in Murach text
- Summary queries
  - GROUP BY and HAVING keywords
  - Aggregate functions: SUM, COUNT, AVG, MIN, MAX

Summary Query with COUNT() and SUM()

```sql
SELECT invoice_id, (invoice_total - payment_total - credit_total) AS balance_due
FROM invoices
WHERE invoice_total - payment_total - credit_total > 0;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

invoice_id	balance_due
89	85.31
94	52.25
98	579.42
99	59.97
100	67.92
101	30.75
102	19351.18
105	503.20
110	90.36
112	10976.06
113	224.00

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

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

number_invoices	total_due
11	32020.42

Examples of Aggregate Functions

```
12  SELECT "after 1/1/2014" AS selection_date,
13          COUNT(*) AS number_invoices,
14          ROUND(AVG(invoice_total),2) AS avg_invoice,
15          MAX(invoice_total) AS highest_invoice,
16          SUM(invoice_total) AS total_invoice,
17          COUNT( DISTINCT vendor_id) AS number_vendors
18  FROM invoices
19  WHERE invoice_date > '2014-01-01';
```

sult Grid | Filter Rows: | Export: | Wrap Cell Content:

selection_date	number_invoices	avg_invoice	highest_invoice	total_invoice	number_vendors
after 1/1/2014	114	1879.74	37966.19	214290.51	34

COUNT with Null Values

```
SELECT invoice_id, invoice_date, payment_date
FROM invoices;
```

Result Grid | Filter Rows: | Edit: | Export/Import:

invoice_id	invoice_date	payment_date
85	2018-07-07	2018-08-11
86	2018-07-07	2018-08-06
87	2018-07-08	2018-08-09
88	2018-07-08	2018-08-11
89	2018-07-10	NULL
90	2018-07-12	2018-08-11
91	2018-07-15	2018-08-06
92	2018-07-15	2018-08-14
93	2018-07-16	2018-08-11
94	2018-07-18	NULL
95	2018-07-19	2018-08-13
96	2018-07-19	2018-08-20
97	2018-07-20	2018-08-07
98	2018-07-21	NULL
99	2018-07-21	NULL
100	2018-07-22	NULL
101	2018-07-22	NULL
102	2018-07-23	NULL

```
SELECT COUNT(invoice_date) AS total_invoices, COUNT(payment_date) AS paid_invoices
FROM invoices
;
```

ult Grid | Filter Rows: | Export: | Wrap Cell Content:

total_invoices	paid_invoices
114	103

GROUP BY Clause

```
36 •  SELECT COUNT(*) AS number_invoices
37    FROM invoices;
```

Result Grid | Filter Rows: | Export:

	number_invoices
▶	114

```
40 •  SELECT vendor_id, COUNT(*) AS number_invoices
41    FROM invoices
42    GROUP BY vendor_id
43    ORDER BY number_invoices DESC;
44
```

sult Grid | Filter Rows: | Export: | Wrap C

vendor_id	number_invoices
123	47
122	9
121	8
95	6
110	5
115	4
37	3
83	2
97	2
34	2
72	2
80	2

GROUP BY Clause WITH ROLLUP

```
51 •  SELECT COUNT(*) AS number_vendors
52    FROM vendors;
53
```

esult Grid | Filter Rows: | Expo

number_vendors
122

```
54 •  SELECT vendor_state, COUNT(*) AS number_vendors
55    FROM vendors
56    GROUP BY vendor_state WITH ROLLUP;
```

esult Grid | Filter Rows: | Export: | Wrap Cell

vendor_state	number_vendors
MA	3
MI	3
MN	1
MO	2
NC	1
NJ	4
NV	2
NY	3
OH	7
PA	3
TN	1
TX	1
VA	1
WI	1
NULL	122

```
58 •  SELECT vendor_state, vendor_city, COUNT(*) AS number_vendors
59    FROM vendors
60    GROUP BY vendor_state, vendor_city WITH ROLLUP;
```

sult Grid | Filter Rows: | Export: | Wrap Cell Content:

vendor_state	vendor_city	number_vendors
NY	Tarrytown	1
NY	NULL	3
OH	Cincinnati	2
OH	Cleves	1
OH	Columbus	2
OH	Marion	1
OH	Oberlin	1
OH	NULL	7
PA	Fort Washington	1
PA	Philadelphia	2
PA	NULL	3
TN	Memphis	1
TN	NULL	1
TX	Dallas	1
TX	NULL	1
VA	Mclean	1
VA	NULL	1
WI	Madison	1
WI	NULL	1
NULL	NULL	122

GROUP BY with Join

```
SELECT vendor_state, vendor_city,
    COUNT(*) AS number_invoice, count(DISTINCT invoices.vendor_id) AS number_vendor
FROM invoices JOIN vendors
    ON invoices.vendor_id = vendors.vendor_id
GROUP BY vendor_state, vendor_city
ORDER BY vendor_state, vendor_city;
```

sult Grid | Filter Rows: | Export: | Wrap Cell Content:

vendor_state	vendor_city	number_invoice	number_vendor
AZ	Phoenix	1	1
CA	Fresno	19	12
CA	Los Angeles	1	1
CA	Oxnard	3	1
CA	Pasadena	5	2
CA	Sacramento	7	2
CA	San Francisco	3	2
CA	Turlock	1	1
CA	Valencia	1	1
DC	Washington	1	1
MA	Boston	1	1
MI	Ann Arbor	5	1
MI	Auburn Hills	2	1

GROUP BY with HAVING Clause

```
73  SELECT vendor_state, vendor_city,
74      COUNT(*) AS number_invoice, count(DISTINCT invoices.vendor_id) AS number_vendor
75  FROM invoices JOIN vendors
76      ON invoices.vendor_id = vendors.vendor_id
77  GROUP BY vendor_state, vendor_city
78  HAVING COUNT(DISTINCT invoices.vendor_id) > 1
79  ORDER BY vendor_state, vendor_city;
```

esult Grid | Filter Rows: | Export: | Wrap Cell Content:

vendor_state	vendor_city	number_invoice	number_vendor
CA	Fresno	19	12
CA	Pasadena	5	2
CA	Sacramento	7	2
CA	San Francisco	3	2

WHERE vs. HAVING

- WHERE filters are applied *before* rows are grouped and aggregates calculated
  - WHERE filters cannot include aggregate functions
  - WHERE filters can contain any column from the base tables

- HAVING filters are applied *after* rows are groups and aggregates calculated
  - HAVING filters can include aggregate functions (SUM, AVG, etc.)
  - HAVING filters can only refer to columns included in the SELECT clause

WHERE vs. HAVING

```sql
SELECT vendor_name, COUNT(*) AS number_invoices, ROUND(AVG(invoice_total),2) AS avg_invoice
FROM vendors JOIN invoices
    ON vendors.vendor_id = invoices.vendor_id
GROUP BY vendors.vendor_id
HAVING AVG(invoice_total) > 500
ORDER BY number_invoices DESC
```

sult Grid | Filter Rows: | Export: | Wrap Cell Content:

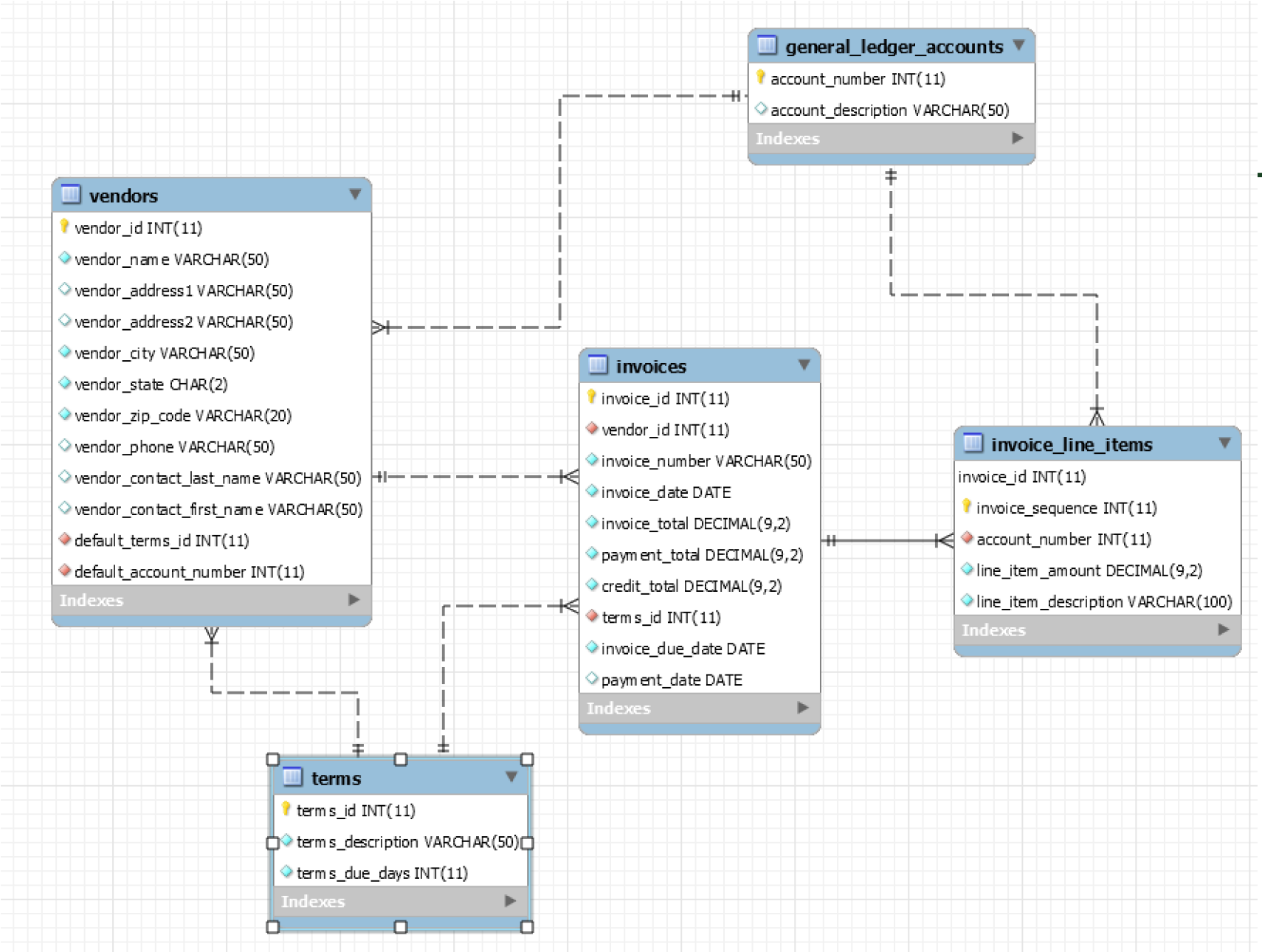
vendor_name	number_invoices	avg_invoice
United Parcel Service	9	2575.33
Zylka Design	8	867.53
Malloy Lithographing Inc	5	23978.48

```sql
SELECT vendor_name, COUNT(*) AS number_invoices, ROUND(AVG(invoice_total),2) AS avg_invoice
FROM vendors JOIN invoices
    ON vendors.vendor_id = invoices.vendor_id
WHERE invoice_total > 500
GROUP BY vendors.vendor_id
ORDER BY number_invoices DESC
;
```

sult Grid | Filter Rows: | Export: | Wrap Cell Content:

vendor_name	number_invoices	avg_invoice
United Parcel Service	9	2575.33
Zylka Design	7	946.67
Malloy Lithographing Inc	5	23978.48

ap database


general_ledger_accounts
account_number INT(11)
account_description VARCHAR(50)
Indexes
vendors
vendor_id INT(11)
vendor_name VARCHAR(50)
vendor_address1 VARCHAR(50)
vendor_address2 VARCHAR(50)
vendor_city VARCHAR(50)
vendor_state CHAR(2)
vendor_zip_code VARCHAR(20)
vendor_phone VARCHAR(50)
vendor_contact_last_name VARCHAR(50)
vendor_contact_first_name VARCHAR(50)
default_terms_id INT(11)
default_account_number INT(11)
Indexes
invoices
invoice_id INT(11)
vendor_id INT(11)
invoice_number VARCHAR(50)
invoice_date DATE
invoice_total DECIMAL(9,2)
payment_total DECIMAL(9,2)
credit_total DECIMAL(9,2)
terms_id INT(11)
invoice_due_date DATE
payment_date DATE
Indexes
invoice_line_items
invoice_id INT(11)
invoice_sequence INT(11)
account_number INT(11)
line_item_amount DECIMAL(9,2)
line_item_description VARCHAR(100)
Indexes
terms
terms_id INT(11)
terms_description VARCHAR(50)
terms_due_days INT(11)
Indexes

Example 1

- Write a SELECT statement that produces a list of each general ledger account and the three columns described below.
  - The GL account description
  - A count of all invoice line items that reference the account number
  - The sum of the invoice line item amount columns that reference the account number
- Return only account numbers that have more than 1 associated invoice line item.
- Sort the list in descending sequence by the sum of the line item amounts.

Example 1

account_description	number_line_items	total_amount
Book Printing Costs	8	148759.97
Freight	60	27599.65
Outside Services	3	13394.10
Book Production Costs	8	6175.12
Books, Dues, and Subscriptions	6	5207.32
Direct Mail Advertising	6	3900.77
Computer Equipment	3	2137.05
Group Insurance	3	564.00
Telephone	7	266.01
Office Supplies	3	175.80

Example 2

- Modify the SELECT statement in Example 1 so that it returns only invoices from Q2 2018 (April 1, 2018 – June 30, 2018).

account_description	number_line_items	total_amount
Book Printing Costs	3	66748.44
Freight	41	17624.19
Outside Services	3	13394.10
Book Production Costs	7	5174.66
Books, Dues, and Subscriptions	4	4027.90
Direct Mail Advertising	5	3810.41
Computer Equipment	3	2137.05
Group Insurance	2	340.00
Telephone	5	193.54
Office Supplies	3	175.80