Lecture 2: Beginning SQL

Monday, January 25, 2015

Agenda for today

- Class Demo: Oracle set up for Mac using Virtual Box (Yuming)
- Brief overview of relational database systems
- SQL: Chapter 3 in Murach textbook

Market shares

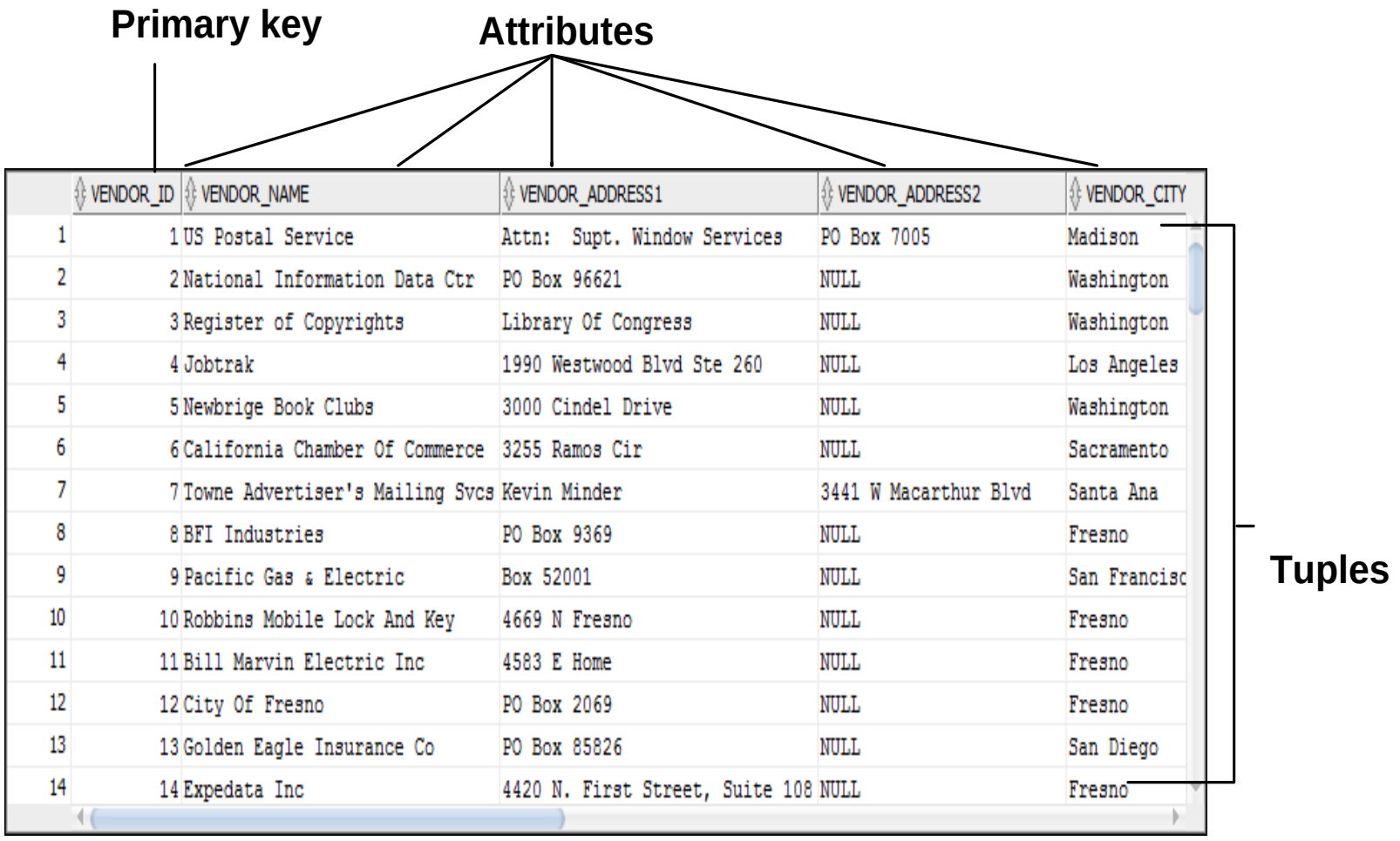
From 2011 Gartner report:

- Oracle: 48% market with $11.7BN in sales
- IBM: 20% market with $4.8BN in sales
- Microsoft: 17% market with $4.0BN in sales
- Other Vendors: 5.8% market with $1.3BN in sales

Basic concepts

- Relational model
- Relation / Entity / Table
- Field / Attribute / Column
- Row / Tuple / Record
- Cell / Value
- Primary key
- Composite primary key
- Foreign key
- Constraint

The Vendors table in the Accounts Payable schema

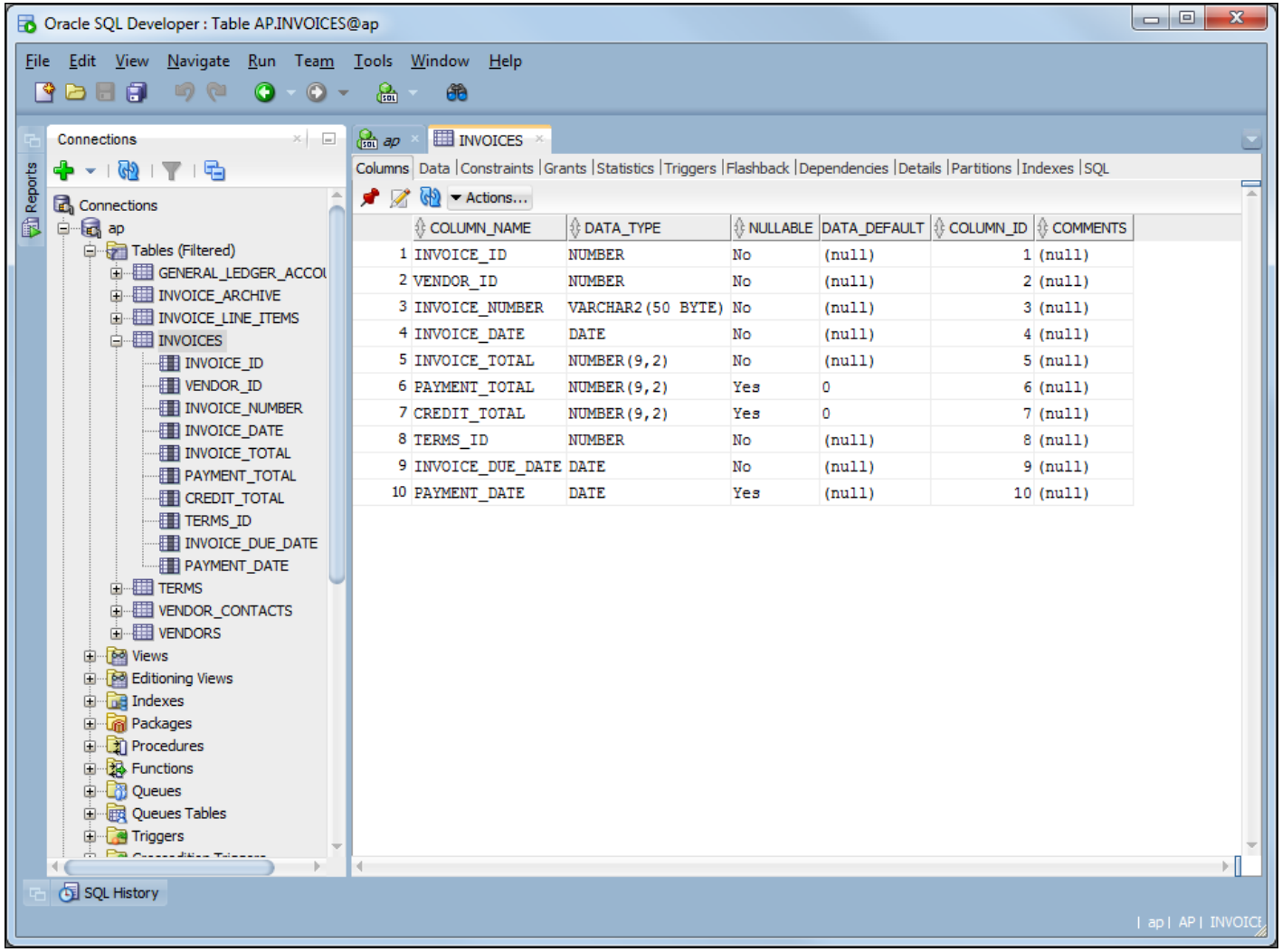


	VENDOR_ID	VENDOR_NAME	VENDOR_ADDRESS1	VENDOR_ADDRESS2	VENDOR_CITY
1	1	US Postal Service	Attn: Supt. Window Services	PO Box 7005	Madison
2	2	National Information Data Ctr	PO Box 96621	NULL	Washington
3	3	Register of Copyrights	Library Of Congress	NULL	Washington
4	4	Jobtrak	1990 Westwood Blvd Ste 260	NULL	Los Angeles
5	5	Newbrige Book Clubs	3000 Cindel Drive	NULL	Washington
6	6	California Chamber Of Commerce	3255 Ramos Cir	NULL	Sacramento
7	7	Towne Advertiser's Mailing Svcs	Kevin Minder	3441 W Macarthur Blvd	Santa Ana
8	8	BFI Industries	PO Box 9369	NULL	Fresno
9	9	Pacific Gas & Electric	Box 52001	NULL	San Francisc
10	10	Robbins Mobile Lock And Key	4669 N Fresno	NULL	Fresno
11	11	Bill Marvin Electric Inc	4583 E Home	NULL	Fresno
12	12	City Of Fresno	PO Box 2069	NULL	Fresno
13	13	Golden Eagle Insurance Co	PO Box 85826	NULL	San Diego
14	14	Expedata Inc	4420 N. First Street, Suite 108	NULL	Fresno

Tables Explained

- A tuple = a record
- A table = a set of records
- The schema of a table is the table name and attributes
- A key is an attribute whose value is unique (by convention, we underline the key)

The columns of the Invoices table



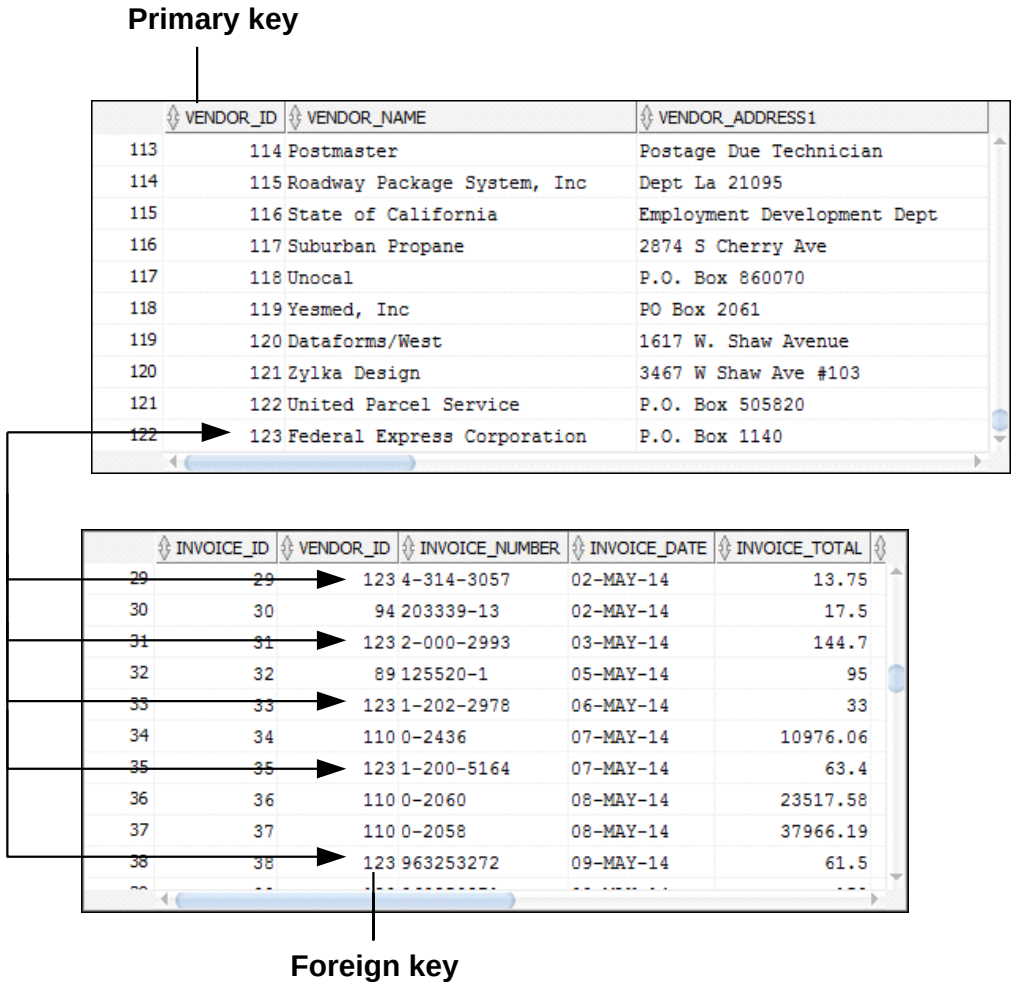
Common data types

- CHAR, VARCHAR2
- NUMBER
- FLOAT
- DATE
- BLOB, CLOB

Constraint types

- NOT NULL constraint
- Unique constraint
- Primary and foreign key constraint
- Check constraint

The relationship between two tables



Relationships between tables

- One-to-many relationship
- One-to-one relationship
- Many-to-many relationship

Principle of data independence

- Physical data independence
- Logical data independence

Examples:

- Adding / dropping a column
- Adding / dropping an index

SQL Introduction

Standard language for querying and manipulating data

Structured **Q**uery **L**anguage

Many standards out there:
- ANSI SQL
- SQL92 (a.k.a. SQL2)
- SQL99 (a.k.a. SQL3)
- Vendors support various subsets of these
- What we discuss is common to all of them

Data Manipulation Language (DML) statements

- SELECT
- INSERT
- UPDATE
- DELETE

Data Definition Language (DDL) statements

- CREATE USER, TABLE, SEQUENCE, INDEX
- ALTER USER, TABLE, SEQUENCE, INDEX
- DROP USER, TABLE, SEQUENCE, INDEX
- GRANT
- REVOKE

The simplified syntax of the SELECT statement

```
SELECT select_list
FROM table_source
[WHERE search_condition]
[ORDER BY order_by_list]
```

The four clauses of the SELECT statement

- SELECT
- FROM
- WHERE
- ORDER BY

A simple SELECT statement

```
SELECT *
FROM invoices
```

	INVOICE_ID	VENDOR_ID	INVOICE_NUMBER	INVOICE_DATE	INVOICE_TOTAL	PAYMENT_TOTAL	CREDIT_TOTAL
1	1	34	QP58872	25-FEB-14	116.54	116.54	0
2	2	34	Q545443	14-MAR-14	1083.58	1083.58	0
3	3	110	P-0608	11-APR-14	20551.18	0	1200
4	4	110	P-0259	16-APR-14	26881.4	26881.4	0

```
(114 rows selected)
```

A SELECT statement that projects and sorts

```
SELECT invoice_number, invoice_date, invoice_total
FROM invoices
ORDER BY invoice_total
```

	INVOICE_NUMBER	INVOICE_DATE	INVOICE_TOTAL
1	25022117	24-MAY-14	6
2	24863706	27-MAY-14	6
3	24780512	29-MAY-14	6
4	21-4748363	09-MAY-14	9.95

```
(114 rows selected)
```

A SELECT statement with no duplicate rows

```
SELECT DISTINCT vendor_city, vendor_state
FROM vendors
ORDER BY vendor_city
```

	VENDOR_CITY	VENDOR_STATE
1	Anaheim	CA
2	Ann Arbor	MI
3	Auburn Hills	MI
4	Boston	MA
5	Brea	CA

```
(53 rows selected)
```

A SELECT statement that retrieves a calculated value

```
SELECT invoice_id, invoice_total,
       (credit_total + payment_total) AS total_credits
FROM invoices
WHERE invoice_id = 17
```

	INVOICE_ID	INVOICE_TOTAL	TOTAL_CREDITS
1	17	356.48	356.48