Intro to Database

Danny Tan

Lecture 1

10/24/19

What is a Database?

- Organized collection of data

e.g

- Price data
- Position data
- Customer data

Way to store Data

- Flat files, CSV, excel, etc
- MS SQL, Oracle, Sybase

Why not just use flat files?

- Flat files reading is slow
- File filtering, manipulating requires time
- Excel has a limitation on number of data

What is SQL

- Structured Query Language
- Special programming language
- Designed for managing data in database

Basic SELECT Statement

SELECT {*, *column* [alias], …}
FROM *table*

- Select all columns use *
- To suppresses duplicates use DISTINCT
- To give selected columns different headings use alias
- If alias have space in them use “ ” , ex “Customer Table”

Example

c_table

Customer	Customer_ID	Address	Last_Order_Date
John A	C_001	100 main street	10/1/2001
Craig B	C_002	101 main street	10/2/2010
Cindy P	C_003	102 main street	11/3/2011
David R	C_004	103 main street	11/4/2011

Example Continued

Select Customer from c_table

Output would be

Customer
John A
Craig B
Cindy P
David R

Using Operators

+ - * /

- Multiplication and division take priority over addition and subtraction

Using the WHERE clause

SELECT {*, *column* [alias], …}

FROM *table*

[WHERE *condition*]

- Strings and dates must be enclosed in single quotation marks (‘ ‘)
- Might be case sensitive depend on the type of database
- More than 1 condition use AND, OR

Example

```
Select * from c_table
where Last_Order_Date <   ‘1/1/2011’
```

Ouput:

Customer	Customer_ID	Address	Last_Order_Date
John A	C_001	100 main street	10/1/2001
Craig B	C_002	101 main street	10/2/2010

Example

stock

stock	trd_date	trd_price
AAPL	11/2/2011	50
TEVA	11/2/2011	30
GE	11/2/2011	20
A	11/2/2011	40

Example

Select trd_price * 2 as price from stock

price
100
60
40
80

Example

What happenes if you don’t add the “as”

(No column name)
100
60
40
80

Comparison Operators

- = : must be exactly the same
- >, >=, <, <=
- <> : not equal to
- BETWEEN… AND… : between 2 values (inclusive)
- IN (list): match any of a list of value
- LIKE: match a character pattern. % represents any sequence of 0 or more characters.

Use NOT if you want to return the opposite value

Examples

- Select * from stock
  where price >= 40
- Select * from stock
  where price between 20 and 30
- Select * from stock
  where stock like ‘%aa%’
- Select * from stock
  where stock like ‘%aa’
- Select * from stock
  where stock like ‘aa%’

Rule of Precedence

All comparison operators → AND → OR

Example:

```
SELECT *
FROM SV
WHERE C_CATEGORY = 'Curve'
OR C_CATEGORY = 'SWVol'
AND C_SYMBOL = ‘USD’
```

Condition1: C_CATEGORY = 'SWVol ‘ AND C_SYMBOL = ‘USD’

Condition2: C_CATEGORY = 'Curve'

Example

stock

stock	trd_date	trd_price	category
AAPL	11/2/2011	50	equity
TEVA	11/2/2011	30	equity
GE	11/2/2011	20	bond
A	11/5/2011	40	bond

Example

```
Select * from stock
where category =   ‘equity’
Or category =   ‘bond’
And  price >= 40
```

Example

stock	trd_date	trd_price	category
AAPL	11/2/2011	50	equity
TEVA	11/2/2011	30	equity
A	11/5/2011	40	bond

Rule of Precedence 2

All comparison operators → AND → OR

Example:

```
SELECT *
FROM SV
WHERE (C_CATEGORY = 'Curve'
OR C_CATEGORY = 'SWVol ‘)
AND C_SYMBOL = ‘USD’
```

Condition1: C_CATEGORY = 'Curve ‘ OR C_CATEGORY = 'SWVol ‘

Condition2: C_SYMBOL = ‘USD ‘

Example

```
Select * from stock
where (category =   ‘equity’
Or price >= 40)
And  trd_date =   ‘11/2/2011
’
```

Example

stock	trd_date	trd_price	category
AAPL	11/2/2011	50	equity
TEVA	11/2/2011	30	equity

ORDER BY Clause

```
SELECT  {*, column [alias], …}
FROM table
[WHERE condition
ORDER BY {column} [ASC|DESC]]
```

- ASC: ascending (this is the default order)
- DESC: descending order
- Null values are displayed last for ascending sequences and first for descending sequences

Example

```
Select * from stock
Order by trd_date
```

stock	trd_date	trd_price	category
AAPL	11/2/2011	50	equity
TEVA	11/2/2011	30	equity
GE	11/2/2011	20	bond
A	11/5/2011	40	bond

Example

```
Select * from stock
Order by trd_date desc
```

stock	trd_date	trd_price	category
A	11/5/2011	40	bond
AAPL	11/2/2011	50	equity
TEVA	11/2/2011	30	equity
GE	11/2/2011	20	bond

Useful functions

- LOWER, UPPER
- RTRIM, LTRIM
- LEN: length of the column
- SUBSTRING(column, start at, length)
- ROUND(column, number of decimal)

Functions may be different in different type of database