Topic C

SQL QUERY #1
Select
Insert/Update/Delete

SQL QUERY I (Select/Insert/Update/Delete)

CONTENT

- Database update
- Retrieve from and populating table
- Insert a row into a table
- Retrieving specific columns
- Removing duplicates : DISTINCT Operator
- Change column label or heading: Alias
- Sorting Result: ORDER BY Clause
- Retrieving Specific Rows: WHERE Clause

Database Update – Update all rows

Product_code	Description	Preimum	Waiting_list	Product_Type	Selling_price	Qty_on_hand
HCH812	SALE HCH812 BLOUSE	N	N	BLOUSE	13.10	20
HG9123	SALE HG9123 BLOUSE	N	Y	BLOUSE	16.40	0

Product

All blouses discount 10%!

Syntax	Question
UPDATE	What do you want to do?
<Table>	What do you want to update?
SET ColumnName1 = ColumnValue1 [,ColumnName2 = ColumnValue2…]	What are the changes?
[WHERE row selection criteria]	What is the condition?

The [ ] indicates that the portion of the statement is optional and '…' indicates that there maybe more such pairs of 'ColumnName = ColumnValue'.

Database update - Update all rows

Product_code	Description	Preimum	Waiting_list	Product_Type	Selling_price	Qty_on_hand
HCH812	SALE HCH812 BLOUSE	N	N	BLOUSE	13.10	20
HG9123	SALE HG9123 BLOUSE	N	Y	BLOUSE	16.40	0

Product

SQL	Syntax	Question
UPDATE	UPDATE	What do you want to do?
Product	<Table>	What do you want to update?
SET Selling_price = Selling_price * 0.9	SET ColumnName1 = ColumnValue1 [,ColumnName2 = ColumnValue2…]	What are the changes?
	[WHERE row selection criteria]	What is the condition?

Product_code	Description	Preimum	Waiting_list	Product_Type	Selling_price	Qty_on_hand
HCH812	SALE HCH812 BLOUSE	N	N	BLOUSE	11.79	20
HG9123	SALE HG9123 BLOUSE	N	Y	BLOUSE	14.76	0

Product (Result set)

SQL Query - Update specific rows

HCH812 - 20% discount !

Product_code	Description	Preimum	Waiting_list	Product_Type	Selling_price	Qty_on_hand
HCH812	SALE HCH812 BLOUSE	N	N	BLOUSE	11.79	20
HG9123	SALE HG9123 BLOUSE	N	Y	BLOUSE	14.76	0

Product

SQL	Syntax	Question
UPDATE	UPDATE	What do you want to do?
Product	<Table>	What do you want to update?
SET Selling_price = Selling_price * 0.8	SET ColumnName1 = ColumnValue1 [,ColumnName2 = ColumnValue2…]	What are the changes?
WHERE Product_code = 'HCH812'	[WHERE row selection criteria]	What is the condition?

Product_code	Description	Preimum	Waiting_list	Product_Type	Selling_price	Qty_on_hand
HCH812	SALE HCH812 BLOUSE	N	N	BLOUSE	9.43	20
HG9123	SALE HG9123 BLOUSE	N	Y	BLOUSE	14.76	0

Product (Result set)

Database update - Update specific rows, multiple columns

HCH812 – 40% DISCOUNT NOW !

LAST 3 PIECES !

Selling_price and Qty_on_hand must be **NUMERIC**, if it meant to use for computation.

Product_code	Description	Preimum	Waiting_list	Product_Type	Selling_price	Qty_on_hand
HCH812	SALE HCH812 BLOUSE	N	N	BLOUSE	9.43	20
HG9123	SALE HG9123 BLOUSE	N	Y	BLOUSE	14.76	0

Product

SQL	Syntax	Question
UPDATE	UPDATE	What do you want to do?
Product	<Table>	What do you want to update?
SET Selling_price = Selling_price * 0.6, Qty_on_hand = 3	SET ColumnName1 = ColumnValue1 [,ColumnName2 = ColumnValue2…]	What are the changes?
WHERE Product_code = 'HCH812'	[WHERE row selection criteria]	What is the condition?

Product_code	Description	Preimum	Waiting_list	Product_Type	Selling_price	Qty_on_hand
HCH812	SALE HCH812 BLOUSE	N	N	BLOUSE	5.66	3
HG9123	SALE HG9123 BLOUSE	N	Y	BLOUSE	14.76	0

Product (Result set)

Database Update - Delete specific rows

SOLD OUT ! HCH812 !

Product_code	Description	Preimum	Waiting_list	Product_Type	Selling_price	Qty_on_hand
HCH812	SALE HCH812 BLOUSE	N	N	BLOUSE	5.66	3
HG9123	SALE HG9123 BLOUSE	N	Y	BLOUSE	14.76	0

Product

SQL	Syntax	Question
DELETE FROM	DELETE FROM	What do you want to do?
Product	<Table>	What do you want to delete?
WHERE Product_code = 'HCH812'	[WHERE row selection criteria]	What is the condition?

Product_code	Description	Preimum	Waiting_list	Product_Type	Selling_price	Qty_on_hand
HG9123	SALE HG9123 BLOUSE	N	Y	BLOUSE	14.76	0

Product (Result set)

Database Update - Delete all rows

Product_code	Description	Preimum	Waiting_list	Product_Type	Selling_price	Qty_on_hand
HG9123	SALE HG9123 BLOUSE	N	Y	BLOUSE	14.76	0

Product

SQL	Syntax	Question
DELETE FROM	DELETE FROM	What do you want to do?
Product	<Table>	What do you want to delete?

Product_code	Description	Preimum	Waiting_list	Product_Type	Selling_price	Qty_on_hand

Product (Result set)

Retrieve from and populating table - retrieve all the data (Method 1)

Title	First_name	Last_name	Email (PK)	Password	DOB
Ms	Linda	See	LindaS@hotmail.com	Abc123	NULL
Mr	David	Lee	DavidL@gmail.com	Da12lee	13/09/1997
Ms	Linda	Soh	LSoh@hotmail.com	Abc123	20/12/1998

Customer

Syntax	Question
SELECT	What do you want to do?
(<Column List>)	What are the columns to display?
FROM	
<Table>	Which table to display?

SQL commands

Command	Syntax
SELECT	SELECT
Title, First_name, Last_name, Email Password, DOB	(<Column List>)
FROM	FROM
Customer	<Table>

Retrieve from and populating table - retrieve all the data (Method 2)

Title	First_name	Last_name	Email	Password	DOB
Ms	Linda	See	LindaS@hotmail.com	Abc123	NULL
Mr	David	Lee	DavidL@gmail.com	Da12lee	13/09/1997
Ms	Linda	Soh	LSoh@hotmail.com	Abc123	20/12/1998

Customer

Syntax	Question
SELECT *	What do you want to do?
FROM	
<Table>	Which table to display?

SQL commands

Command	Syntax
SELECT	SELECT
*	(<Column List>)
FROM	FROM
Customer	<Table>

How to insert a Row Into a Table?

Column name	Data Type	Null Constraint
Title	Char(2)	NOT NULL
First_name	Varchar(20)	NOT NULL
Last_name	Varchar(20)	NOT NULL
Email	Varchar(50)	NOT NULL
Password	Char(8)	NOT NULL
DOB	Date	NULL

Customer

Syntax	Question
INSERT INTO	What do you want to do?
<Table name>	What is the table name?
(<Column List>)	What are the columns to insert?
VALUES	
(<Column value list>)	What are the values to insert for each column?

SQL commands

Command	Syntax
INSERT INTO CUSTOMER	INSERT INTO <Table name>
(Title, First_name, Last_name, Email, Password, DOB)	(<Column List>)
VALUES	VALUES
('Ms', 'Linda', 'See', 'LindaS@Hotmail.com, 'Abc123', NULL)	(<Column value list>)

Insert a row into a table - Column List and Column Value List

```
INSERT INTO CUSTOMER
(Title, First_name, Last_name, Email, Password, DOB)
VALUES
('Ms', 'Linda', 'See', 'LindaS@hotmail.com', 'Abc123', NULL)
```

- The number of items in each list must match the values.

```
INSERT INTO CUSTOMER
(Title, First_name, Last_name, Email, Password, DOB)
VALUES
('Ms', 'Linda', 'LindaS@hotmail.com', 'See', 'Abc123', NULL)
```

- Direct correspondence in the positions of items in the column value list and the column list.

Insert a row into a table - Column List and Column Value List

```
INSERT INTO CUSTOMER
(Email, First_name, Last_name, Title, Password, DOB)
VALUES
('LindaS@hotmail.com', 'Linda', 'See', 'Ms', 'Abc123', NULL)
```

Customer

Column name	Data Type	Null Constraint
Title	Char(2)	NOT NULL
First_name	Varchar(20)	NOT NULL
Last_name	Varchar(20)	NOT NULL
Email	Varchar(50)	NOT NULL
Password	Char(8)	NOT NULL
DOB	Date	NULL

- The positions of the items do not need to be the same as the table.

Insert a row into a table - Column List and Column Value List

```
INSERT INTO CUSTOMER
(Email, First_name, Last_name, Title, Password, DOB)
VALUES
('LindaS@hotmail.com', 'Linda', 'See', 'Ms', 'Abc123', '1995_12_25')
```

Customer

Column name	Data Type	Null Constraint
Title	Char(2)	NOT NULL
First_name	Varchar(20)	NOT NULL
Last_name	Varchar(20)	NOT NULL
Email	Varchar(50)	NOT NULL
Password	Char(8)	NOT NULL
DOB	Date	NULL

- The data type for each corresponding items in the list must be compatible.

Insert a row into a table - Column List and Column Value List

```
INSERT INTO CUSTOMER
(Title, First_name, Last_name, Email, Password, DOB)
VALUES
('Mr', 'David', 'Lee', 'DavidL@hotmail.com', 'Da12lee', '09/13/1997')
```

- Character literals and date input **must be enclosed** in **single quotes**. Date format is **mm/dd/yyyy (by default)**.

```
INSERT INTO CART
(Email, Product, Size, Ref, Qty)
VALUES
('DavidL@Hotmail.com', 'HG4872' NULL, NULL 1)
```

- Numeric and NULL column values **must not be enclosed** in **single quotes**.

Insert a row into a table - Column List and Column Value List

```
INSERT INTO CUSTOMER
(Title, First_name, Last_name, Email, Password, DOB)
VALUES
('Mr', 'David', 'Lee', 'DavidL@hotmail.com', 'Da12lee', '09/13/1997')
```

```
INSERT INTO CUSTOMER
(Title, First_name, Last_name, Email, Password, DOB)
VALUES
('Mr', 'David', 'Lee', 'davidl@hotmail.com', 'Da12lee', '09/13/1997')
```

```
INSERT INTO CUSTOMER
(Title, First_name, Last_name, Email, Password, DOB)
VALUES
('Mr', 'David', 'Lee', 'DAVIDL@HOTMAIL.COM' 'Da12lee',
'09/13/1997')
```

- Data that is stored in the table is **NOT Case-Sensitive (SQL Server – by default).**

Insert a row into a table - Column List and Column Value List

```
INSERT INTO CUSTOMER
(Title, First_name, Last_name, Email, Password, DOB)
VALUES
('Mr', 'David', 'Lee', 'DavidL@hotmail.com', 'Da12lee', '09/13/1997')
```

INSERT SUCCESSFULLY

```
INSERT INTO CUSTOMER
(Title, First_name, Last_name, Email, Password, DOB)
VALUES
('Mr', 'David', 'Lee', 'davidl@hotmail.com', Da12lee', '09/13/1997')
```

VIOLATION OF PRIMARY KEY CONSTRAINT

```
INSERT INTO CUSTOMER
(Title, First_name, Last_name, Email, Password, DOB)
VALUES
('Mr', 'David', 'Lee', 'DAVIDL@HOTMAIL.COM', 'Da12lee',
'09/13/1997')
```

Insert a row into a table - Column List and Column Value List

```
INSERT INTO CUSTOMER
(Title, First_name, Last_name, Email, Password, DOB)
VALUES
('Mr', 'David''s', 'Lee', 'DavidL@hotmail.com', 'Da12lee', '09/13/1997')
```

SQL server will store the First_name as David's

- To store data with a quote, you have to double the quote.

C

Insert a row into a table - Column List and Column Value List

Column name	Data Type	Null Constraint
Title	Char(2)	NOT NULL
First_name	Varchar(20)	NOT NULL
Last_name	Varchar(20)	NOT NULL
Email	Varchar(50)	NOT NULL
Password	Char(8)	NOT NULL
DOB	Date	NULL

Customer

SQL commands

Syntax	Question
INSERT INTO	What do you want to do?
<Table name>	What is the table name?
VALUES	
(<Column value list>)	What are the values to insert for each column?

Command	Syntax
INSERT INTO CUSTOMER	INSERT INTO <Table name>
VALUES	VALUES
('Ms', 'Linda', 'See', 'LindaS@Hotmail.com, 'Abc123', NULL)	(<Column value list>)

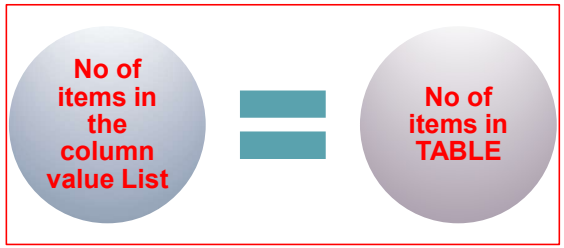
- The column list is OPTIONAL.

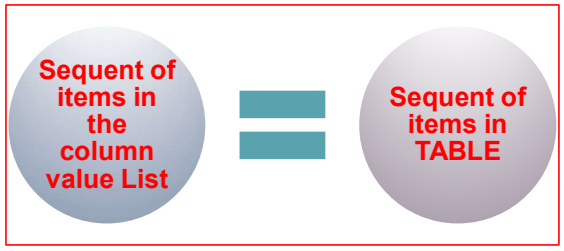
Insert a row into a table - Column List and Column Value List

Customer

Column name	Data Type	Null Constraint
Title	Char(2)	NOT NULL
First_name	Varchar(20)	NOT NULL
Last_name	Varchar(20)	NOT NULL
Email	Varchar(50)	NOT NULL
Password	Char(8)	NOT NULL
DOB	Date	NULL

```
INSERT INTO CUSTOMER
VALUES
('Ms', 'Linda', 'See', 'LindaS@hotmail.com', 'Abc123', NULL)
```



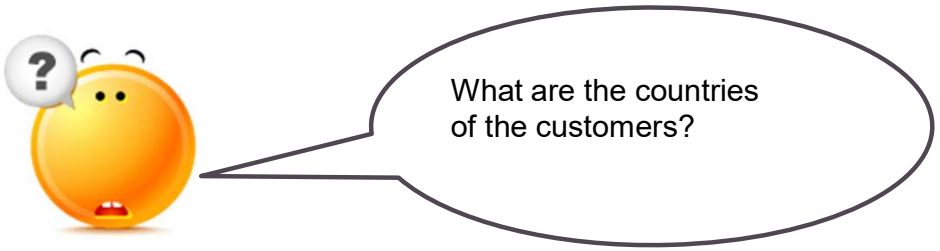


Retrieving specific columns

Address_ID	Address1	Address2	Email	Country	Pcode
My address 1	Blk 145, Toa Payoh,	Lorong 1, #07-12	DavidL@gmail.com	Singapore	145112
My address 2	Blk 123, Ang Mo Kio	Ave 6, #12-12	DavidL@gmail.com	Singapore	123121
Addr1	123 Flora Road	Garden Way	LSoh@hotmail.com	Singapore	503984
A1	22 Lin Hua Road	NULL	LindaS@hotmail.com	China	345982
A	18 Huat Road	NULL	magS@hotmail.com	Malaysia	234562

Address

SELECT Country FROM Address



Resulting Table

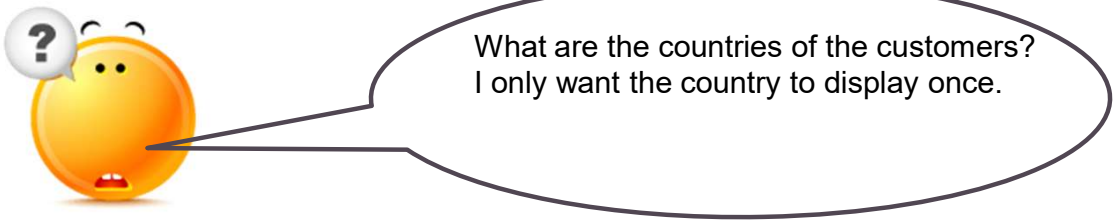
Country
Singapore
Singapore
Singapore
China
Malaysia

Removing duplicates : DISTINCT Operator

Address_ID	Address1	Address2	Email	Country	Pcode
My address 1	Blk 145, Toa Payoh,	Lorong 1, #07-12	DavidL@gmail.com	Singapore	145112
My address 2	Blk 123, Ang Mo Kio	Ave 6, #12-12	DavidL@gmail.com	Singapore	123121
Addr1	123 Flora Road	Garden Way	LSoh@hotmail.com	Singapore	503984
A1	22 Lin Hua Road	NULL	LindaS@hotmail.com	China	345982
A	18 Huat Road	NULL	magS@hotmail.com	Malaysia	234562

Address

SELECT **DISTINCT** Country FROM Address



Country
Singapore
China
Malaysia

Resulting Table

Change column label or heading (ALIAS) using keyword AS

Address_ID	Address1	Address2	Email	Country	Pcode
My address 1	Blk 145, Toa Payoh,	Lorong 1, #07-12	DavidL@gmail.com	Singapore	145112
My address 2	Blk 123, Ang Mo Kio	Ave 6, #12-12	DavidL@gmail.com	Singapore	123121
Addr1	123 Flora Road	Garden Way	LSoh@hotmail.com	Singapore	503984
A1	22 Lin Hua Road	NULL	LindaS@hotmail.com	China	345982
A	18 Huat Road	NULL	magS@hotmail.com	Malaysia	234562

Address

```
SELECT
country
FROM
```

AS

```
email,
'Country Name'
Address
```

List the email and country of each customer. Use 'Country Name' column label or heading.

Email	Country Name
DavidL@gmail.com	Singapore
DavidL@gmail.com	Singapore
LSoh@hotmail.com	Singapore
LindaS@hotmail.com	China
magS@hotmail.com	Malaysia

Resulting Table

Change column label or heading (ALIAS) WITHOUT keyword AS

Address_ID	Address1	Address2	Email	Country	Pcode
My address 1	Blk 145, Toa Payoh,	Lorong 1, #07-12	DavidL@gmail.com	Singapore	145112
My address 2	Blk 123, Ang Mo Kio	Ave 6, #12-12	DavidL@gmail.com	Singapore	123121
Addr1	123 Flora Road	Garden Way	LSoh@hotmail.com	Singapore	503984
A1	22 Lin Hua Road	NULL	LindaS@hotmail.com	China	345982
A	18 Huat Road	NULL	magS@hotmail.com	Malaysia	234562

Address

SELECT country FROM Address **AS** **email, 'Country Name'**

The use of the **AS keyword** is actually **optional**. That is, it can be omitted but maintain the same effect.

Email	Country Name
DavidL@gmail.com	Singapore
DavidL@gmail.com	Singapore
LSoh@hotmail.com	Singapore
LindaS@hotmail.com	China
magS@hotmail.com	Malaysia

Resulting Table

Change column label or heading using underscore in ALIAS

```
SELECT
country
FROM
```

```
email,
Country_Name
Address
```

With underscore

Email	Country _Name
DavidL@gmail.com	Singapore
DavidL@gmail.com	Singapore
LSoh@hotmail.com	Singapore
LindaS@hotmail.com	China
magS@hotmail.com	Malaysia

Resulting Table

```
SELECT
country
FROM
```

```
email,
Country Name
Address
```

Without underscore

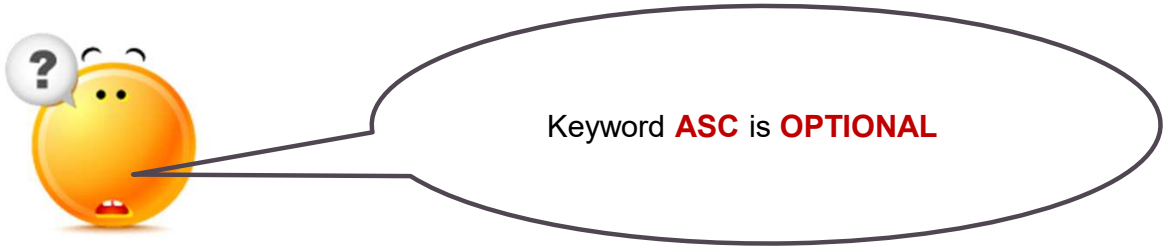
Incorrect syntax near ‘name’

Sorting Resulting (Single Column Sorting)

```
SELECT *
FROM Country
ORDER BY Country ASC
```

Country
Canada
China
Japan
Korea
Malaysia
Singapore
Thailand
USA

Resulting Table



Sorting Resulting (Using Column Number as Shortcut)

Address_ID	Address1	Address2	Email	Country	Pcode
My address 1	Blk 145, Toa Payoh,	Lorong 1, #07-12	DavidL@gmail.com	Singapore	145112
My address 2	Blk 123, Ang Mo Kio	Ave 6, #12-12	DavidL@gmail.com	Singapore	123121
Addr1	123 Flora Road	Garden Way	LSoh@hotmail.com	Singapore	503984
A1	22 Lin Hua Road	NULL	LindaS@hotmail.com	China	345982
A	18 Huat Road	NULL	magS@hotmail.com	Malaysia	234562

Address

Substitute the **column names** with the **column number**

Email	Country
magS@hotmail.com	Malaysia
LSoh@hotmail.com	Singapore
LindaS@hotmail.com	China
DavidL@gmail.com	Singapore
DavidL@gmail.com	Singapore

Resulting Table

Retrieving Specific Rows: WHERE Clause
Search condition using comparison

SQL Comparison Operator	USE
=	Equals
<> Or !=	Is not equal to
<	Is less than
>	Is more than
<=	Is less than or equal to
>=	Is more than or equal to

Retrieving Specific Rows: WHERE Clause
Single Search Condition

Product_Code	Selling_price	Prod_Description	Premium	Waiting_list	Qty_on_hand
HG4953	17.60	Body Pants	Y	NULL	10
RQ8996	22.00	Dress Yellow	NULL	Y	0
DT4732	17.20	Top + Skirt White	Y	NULL	3
RQ7601	26.50	Dress Colour	NULL	NULL	4

Product

List the description all the premium Product and rename the column as 'Premium Product'

SQL	Syntax	Question
SELECT Prod_Description AS 'Premium Product'	SELECT <List of columns>	What do you want to do?
FROM Product	FROM <Table>	What table do you want to use?
WHERE premium = 'y'	WHERE <row selection criteria>	What is the condition?

Premium Product
Body Pants
Top + Skirt White

Resulting Table

Retrieving Specific Rows: WHERE Clause
Single Search Condition

Product_Code	Selling_price	Prod_Description	Premium	Waiting_list	Qty_on_hand
HG4953	17.60	Body Pants	Y	NULL	10
RQ8996	22.00	Dress Yellow	NULL	Y	0
DT4732	17.20	Top + Skirt White	Y	NULL	3
RQ7601	26.50	Dress Colour	NULL	NULL	4

Product

List the product_code, description and selling price of product where the price is more than $20. List in descending order the price.

SQL	Syntax	Question
SELECT product_code, Prod_description, Selling_price	SELECT <List of columns>	What do you want to do?
FROM Product	FROM <Table>	What table do you want to use?
WHERE Selling_price > 20	WHERE <row selection criteria>	What is the condition?
ORDER BY 3 DESC		What is the sorting order?

Product_code	Prod_Description	Selling_price
RQ7601	Dress Colour	26.50
RQ8996	Dress Yellow	22.00

Resulting Table

Retrieving Specific Rows: WHERE Clause
Search condition using Range: Between

List the product code, description and selling px of product where selling px between $22 to $30. List in descending order the price. Provide meaningful names to the column. Include those exactly $22 and $30

Product_Code	Selling_price	Prod_Description	Premium	Waiting_list	Qty_on_hand
HG4953	17.60	Body Pants	Y	NULL	10
RQ8996	22.00	Dress Yellow	NULL	Y	0
DT4732	17.20	Top + Skirt White	Y	NULL	3
RQ7601	26.50	Dress Colour	NULL	NULL	4

Product

SQL	Syntax	Question
SELECT product_code 'Product Code', Prod_description 'Product Description', Selling_price	SELECT <List of columns>	What do you want to do?
FROM Product	FROM <Table>	What table do you want to use?
WHERE Selling_price BETWEEN 22 AND 30	WHERE <row selection criteria>	What is the condition?
ORDER BY 3 DESC		What is the sorting order?

Product Code	Product Description	Selling_price
RQ7601	Dress Colour	26.50
RQ8996	Dress Yellow	22.00

Resulting Table

BETWEEN will INCLUDE both the condition range.
Eg: $22 and $30

Retrieving Specific Rows: WHERE Clause

Search condition using Range: (Option 1)Not Between

Product_Code	Selling_price	Prod_Description	Premium	Waiting_list	Qty_on_hand
HG4953	17.60	Body Pants	Y	NULL	10
RQ8996	22.00	Dress Yellow	NULL	Y	0
DT4732	17.20	Top + Skirt White	Y	NULL	3
RQ7601	26.50	Dress Colour	NULL	NULL	4

List the product code, description and selling price of product where unit px is less than $22 or more than $30. List in descending order the price.

Product

SQL	Syntax	Question
SELECT product_code, prod_description, selling_price	SELECT <List of columns>	What do you want to do?
FROM Product	FROM <Table>	What table do you want to use?
WHERE selling_price NOT BETWEEN 22 AND 30	WHERE <row selection criteria>	What is the condition?
ORDER BY 3 DESC		What is the sorting order?

Product_code	Prod_Description	Selling_price
HG4953	Body Pants	17.60
DT4732	Top + Skirt White	17.20

Resulting Table

NOT BETWEEN will NOT INCLUDE both the condition range. Eg: $22 and $30

Retrieving Specific Rows: WHERE Clause
Search condition using Range: Option 2: using OR

List the product code, description and selling price of product where selling price is less than $22 or more than $30. List in descending order the price.

Product_Code	Selling_price	Prod_Description	Premium	Waiting_list	Qty_on_hand
HG4953	17.60	Body Pants	Y	NULL	10
RQ8996	22.00	Dress Yellow	NULL	Y	0
DT4732	17.20	Top + Skirt White	Y	NULL	3
RQ7601	26.50	Dress Colour	NULL	NULL	4

Product

SQL	Syntax	Question
SELECT product_code, prod_description, selling_price	SELECT <List of columns>	What do you want to do?
FROM Product	FROM <Table>	What table do you want to use?
WHERE selling_price < 22 OR selling_price > 30	WHERE <row selection criteria>	What is the condition?
ORDER BY 3 DESC		What is the sorting order?

Product_code	Prod_Description	Selling_price
HG4953	Body Pants	17.60
DT4732	Top + Skirt White	17.20

Resulting Table

Retrieving Specific Rows: WHERE Clause
Search condition using (Option 1)Set Membership IN

Product_Code	Selling_price	Prod_Description	Premium	Waiting_list	Qty_on_hand
HG4953	17.60	Body Pants	Y	NULL	10
RQ8996	22.00	Dress Yellow	NULL	Y	0
DT4732	17.20	Top + Skirt White	Y	NULL	3
RQ7601	26.50	Dress Colour	NULL	NULL	4

Product

List the product code, description and unit px of product where the product description are 'Body Pants' or 'Dress Colour'. List in Ascending order of description.

SQL	Syntax	Question
SELECT product_code, prod_description, selling_price	SELECT <List of columns>	What do you want to do?
FROM Product	FROM <Table>	What table do you want to use?
WHERE prod_description IN ('Body Pants', 'Dress Colour')	WHERE <row selection criteria>	What is the condition?
ORDER BY 2		What is the sorting order?

Product_code	Prod_Description	Selling_price
HG4953	Body Pants	17.60
RQ7601	Dress Colour	26.50

Resulting Table

IN operator is used to check the Column value with the condition. If it is match, the predicate is TRUE and the row is retrieved.

Retrieving Specific Rows: WHERE Clause
Search condition (Option 2) using OR

Product_Code	Selling_price	Prod_Description	Premium	Waiting_list	Qty_on_hand
HG4953	17.60	Body Pants	Y	NULL	10
RQ8996	22.00	Dress Yellow	NULL	Y	0
DT4732	17.20	Top + Skirt White	Y	NULL	3
RQ7601	26.50	Dress Colour	NULL	NULL	4

Product

List the product code, description and selling price of product where the product description are 'Body Pants' or 'Dress Colour'. List in Ascending order of description.

SQL	Syntax	Question
SELECT product_code, prod_description, selling_price	SELECT <List of columns>	What do you want to do?
FROM Product	FROM <Table>	What table do you want to use?
WHERE prod_description = 'Body Pants' OR prod_description = 'Dress Colour'	WHERE <row selection criteria>	What is the condition?
ORDER BY 2		What is the sorting order?

Product_code	Prod_Description	Selling_price
HG4953	Body Pants	17.60
RQ7601	Dress Colour	26.50

Resulting Table

Retrieving Specific Rows: WHERE Clause
Search condition using Set Membership NOT IN

List the product code, description and unit px of product where the product description are not 'Body Pants' and 'Dress Colour'. List in Ascending order of description.

Product_Code	Selling_price	Prod_Description	Premium	Waiting_list	Qty_on_hand
HG4953	17.60	Body Pants	Y	NULL	10
RQ8996	22.00	Dress Yellow	NULL	Y	0
DT4732	17.20	Top + Skirt White	Y	NULL	3
RQ7601	26.50	Dress Colour	NULL	NULL	4

Product

SQL	Syntax	Question
SELECT product_code, prod_description, selling_price	SELECT <List of columns>	What do you want to do?
FROM Product	FROM <Table>	What table do you want to use?
WHERE prod_description NOT IN ('Body Pants', 'Dress Colour')	WHERE <row selection criteria>	What is the condition?
ORDER BY 2		What is the sorting order?

Resulting Table

Product_code	Prod_Description	Selling_price
RQ8996	Dress Yellow	22.00
DT4732	Top + Skirt White	17.20

NOT IN operator is used to check the Column value with the condition. If it is not match, the predicate is TRUE and the row is retrieved.

Retrieving Specific Rows: WHERE Clause
Search condition using Set Membership LIKE

Product_Code	Selling_price	Prod_Description	Premium	Waiting_list	Qty_on_hand
HG4953	17.60	Body Pants	Y	NULL	10
RQ8996	22.00	Dress Yellow	NULL	Y	0
DT4732	17.20	Top + Skirt White	Y	NULL	3
RQ7601	26.50	Dress Colour	NULL	NULL	4

Product

List the product code, description and selling price of product where the product description start with 'D'. List in Ascending order of description.

SQL	Syntax	Question
SELECT product_code, prod_description, selling_price	SELECT <List of columns>	What do you want to do?
FROM Product	FROM <Table>	What table do you want to use?
WHERE prod_description LIKE 'D%'	WHERE <row selection criteria>	What is the condition?
ORDER BY 2		What is the sorting order?

Product_code	Prod_Description	Selling_price
RQ7601	Dress Colour	26.50
RQ8996	Dress Yellow	22.00

Resulting Table

LIKE operator is used to perform Pattern recognition. % sign represents any sequence of zero or more characters. It is like wildcard. A row is selected if it matches the pattern.

Retrieving Specific Rows: WHERE Clause
Search condition using Set Membership NOT LIKE

Product_Code	Selling_price	Prod_Description	Premium	Waiting_list	Qty_on_hand
HG4953	17.60	Body Pants	Y	NULL	10
RQ8996	22.00	Dress Yellow	NULL	Y	0
DT4732	17.20	Top + Skirt White	Y	NULL	3
RQ7601	26.50	Dress Colour	NULL	NULL	4

Product

List the product code, description and selling price of product where the product description does not start with 'D'. List in Ascending order of description.

SQL	Syntax	Question
SELECT product_code, prod_description, selling_price	SELECT <List of columns>	What do you want to do?
FROM Product	FROM <Table>	What table do you want to use?
WHERE prod_description NOT LIKE 'D%'	WHERE <row selection criteria>	What is the condition?
ORDER BY 2		What is the sorting order?

Product_code	Prod_Description	Selling_price
HG4953	Body Pants	17.60
DT4732	Top + Skirt White	17.20

Resulting Table

NOT LIKE operator is also used to perform Pattern recognition. A row is select if it does not match the pattern.

Retrieving Specific Rows: WHERE Clause
Search condition using Set Membership LIKE

List the product code, description and selling price of product where the product code start with 'R' and end with '8996'. List in Ascending order of description.

Product_Code	Selling_price	Prod_Description	Premium	Waiting_list	Qty_on_hand
HG4953	17.60	Body Pants	Y	NULL	10
RQ8996	22.00	Dress Yellow	NULL	Y	0
DT4732	17.20	Top + Skirt White	Y	NULL	3
RQ7601	26.50	Dress Colour	NULL	NULL	4

Product

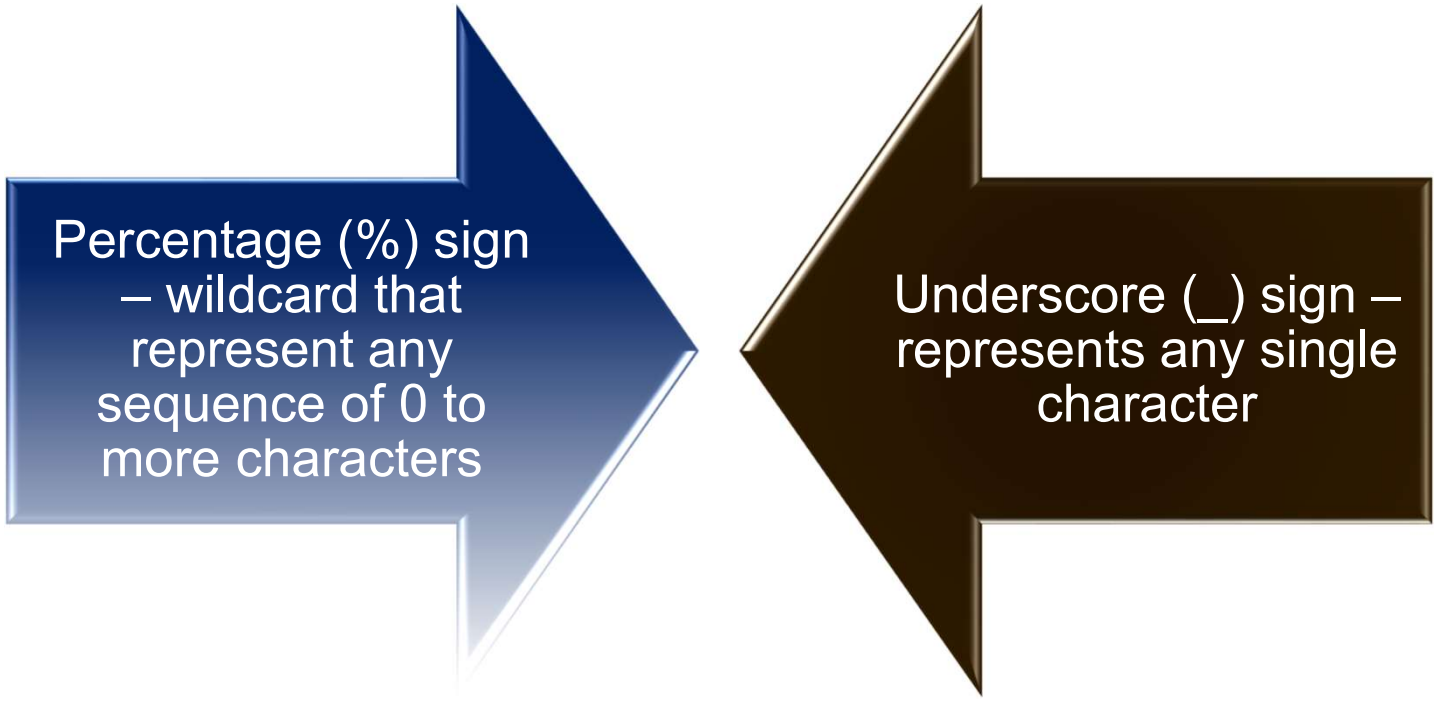
SQL	Syntax	Question
SELECT product_code, prod_description, selling_price	SELECT <List of columns>	What do you want to do?
FROM Product	FROM <Table>	What table do you want to use?
WHERE product_code LIKE 'R_8996'	WHERE <row selection criteria>	What is the condition?
ORDER BY 2		What is the sorting order?

Product_code	Prod_Description	Selling_price
RQ8996	Dress Yellow	22.00

Resulting Table

_ sign is also used to perform Pattern recognition and it represents any single characters. A row is selected if it matches the pattern.

Retrieving Specific Rows: WHERE Clause
Pattern Matching characters



Retrieving Specific Rows: WHERE Clause
Search condition using IS NULL

List the product code, description and selling price of product where premium is NULL. Descending order of description.

Product_Code	Selling_price	Prod_Description	Premium	Waiting_list	Qty_on_hand
HG4953	17.60	Body Pants	Y	NULL	10
RQ8996	22.00	Dress Yellow	NULL	Y	0
DT4732	17.20	Top + Skirt White	Y	NULL	3
RQ7601	26.50	Dress Colour	NULL	NULL	4

Product

SQL	Syntax	Question
SELECT product_code, prod_description, selling_price	SELECT <List of columns>	What do you want to do?
FROM Product	FROM <Table>	What table do you want to use?
WHERE premium IS NULL	WHERE <row selection criteria>	What is the condition?
ORDER BY 2 DESC		What is the sorting order?

Product_code	Description	Premium
RQ8996	Dress Yellow	NULL
RQ7601	Dress Colour	NULL

Resulting Table

IS NULL operator is used to search for the absence of value. A row is selected if it is NULL

Retrieving Specific Rows: WHERE Clause
Search condition using IS NOT NULL

List the product code, description and premium of those premium products. Descending order of description.

Product_Code	Selling_price	Prod_Description	Premium	Waiting_list	Qty_on_hand
HG4953	17.60	Body Pants	Y	NULL	10
RQ8996	22.00	Dress Yellow	NULL	Y	0
DT4732	17.20	Top + Skirt White	Y	NULL	3
RQ7601	26.50	Dress Colour	NULL	NULL	4

Product

SQL	Syntax	Question
SELECT product_code, prod_description, selling_price	SELECT <List of columns>	What do you want to do?
FROM Product	FROM <Table>	What table do you want to use?
WHERE premium IS NOT NULL	WHERE <row selection criteria>	What is the condition?
ORDER BY 2 DESC		What is the sorting order?

Product_code	Prod_Description	Premium
HG4953	Body Pants	Y
DT4732	Top + Skirt	Y

Resulting Table

IS NOT NULL operator is used to search for the present of value. A row is selected if it is NOT NULL

Retrieving Specific Rows: WHERE Clause
Search condition using Logical operator OR

Product

Product_Code	Selling_price	Prod_Description	Premium	Waiting_list	Qty_on_hand
HG4953	17.60	Body Pants	Y	NULL	10
RQ8996	22.00	Dress Yellow	NULL	Y	0
DT4732	17.20	Top + Skirt White	Y	NULL	20
RQ5503	NULL	Dress Colour	NULL	NULL	30
DT9901	NULL	Long Dress	Y	NULL	2

OR CONDITION

Condition A	Condition B	Result
TRUE	TRUE	TRUE
TRUE	FALSE	TRUE
TRUE	NULL	TRUE
FALSE	TRUE	TRUE
FALSE	FALSE	FALSE
FALSE	NULL	NULL
NULL	TRUE	TRUE
NULL	FALSE	NULL
NULL	NULL	NULL

EXAMPLE: WHAT WILL THE FOLLOWING OR CONDITIONS RETURN?

Product_code	Selling_price (condition A)	Premium (condition B)	Resulting table

Retrieving Specific Rows: WHERE Clause
Search condition using Logical operator AND

Product

Product_Code	Unit_px	Description	Premium	Waiting_list
HG4953	17.60	Body Pants	Y	NULL
RQ8996	22.00	Dress Yellow	NULL	Y
DT4732	17.20	Top + Skirt White	Y	NULL
RQ5503	NULL	Dress Colour	NULL	NULL
DT9901	NULL	Long Dress	Y	NULL

AND CONDITION		
Condition A	Condition B	Result
TRUE	TRUE	TRUE
TRUE	FALSE	FALSE
TRUE	NULL	NULL
FALSE	TRUE	FALSE
FALSE	FALSE	FALSE
FALSE	NULL	FALSE
NULL	TRUE	NULL
NULL	FALSE	FALSE
NULL	NULL	NULL

EXAMPLE: WHAT WILL THE FOLLOWING AND CONDITIONS RETURN?

Product_code	Selling_price (condition A)	Premium (condition B)	Resulting table

Retrieving Specific Rows: WHERE Clause
Search condition using Logical operator NOT

Product

Product_Code	Unit_px	Description	Premium	Waiting_list
HG4953	17.60	Body Pants	Y	NULL
RQ8996	22.00	Dress Yellow	NULL	Y
DT4732	17.20	Top + Skirt White	Y	NULL
RQ5503	NULL	Dress Colour	NULL	NULL
DT9901	NULL	Long Dress	Y	NULL

NOT CONDITION		USAGE OF NOT CONDITION
Condition	Result	NOT IN NOT LIKE NOT BETWEEN IS NOT NULL
TRUE	FALSE	
FALSE	TRUE	
NULL	NULL	

NOT is a unary operator and it operates on one condition only