https://unsplash.com/photos/gm3bxHin8VA

Aalto University
School of Business

MySQL for Data Analytics

Lecturer: Yong Liu
Contact me at: Yong.liu@aalto.fi

Content for class 5

- **Union**
- **Group by**
- **Manipulation on NULL value**
- **Manipulation on date**
- **Sub-queries**
- **Alter and update table**
- **Delete record**

Sample Question 1

- You are requested to do an analysis on a data file containing the sales information of your company before 2022. Recently, you received a complementary data file of the sales information for 2022. Both files have the same data structure. You want to merge two tables into one table. How would you do that?

UNION [all]

- **UNION** operator combines two or more result sets from multiple **SELECT statements**
  - Each SELECT statement within UNION must have the same number of columns
  - The corresponding columns must also have similar data types

Solution

Select * from sale_before_2022

Union

select * from sale_in_2022

“Union” removes duplicated rows in default;
“Union all” keeps duplicated rows.

Question 2

classicmodels.customers: 122 row

Next

customerNumber	contactLastName	contactFirstName	phone	country
459	Ottlieb	Sven	0241-039123	Germany
157	Leong	Kelvin	2155551555	USA
303	Schuyler	Bradley	+31 20 491 9555	Netherlands
496	Snowden	Tony	+64 9 5555500	New Zealand
323	Graham	Mike	+64 9 312 5555	New Zealand
357	MacKinlay	Wales	64-9-3763555	New Zealand
216	Saavedra	Eduardo	(93) 203 4555	Spain

Table customer

classicmodels.employees: 23 rows total (approximately)

employeeNumber	lastName	firstName	extension	email
1,143	Bow	Anthony	x5428	abow@classicmodelcars.com
1,076	Firrelli	Jeff	x9273	jfirrelli@classicmodelcars.com
1,165	Jennings	Leslie	x3291	ljennings@classicmodelcars.com
1,002	Murphy	Diane	x5800	dmurphy@classicmodelcars.com
1,056	Patterson	Mary	x4611	mpatterso@classicmodelcars.com
1,166	Thompson	Leslie	x4065	lthompson@classicmodelcars.com

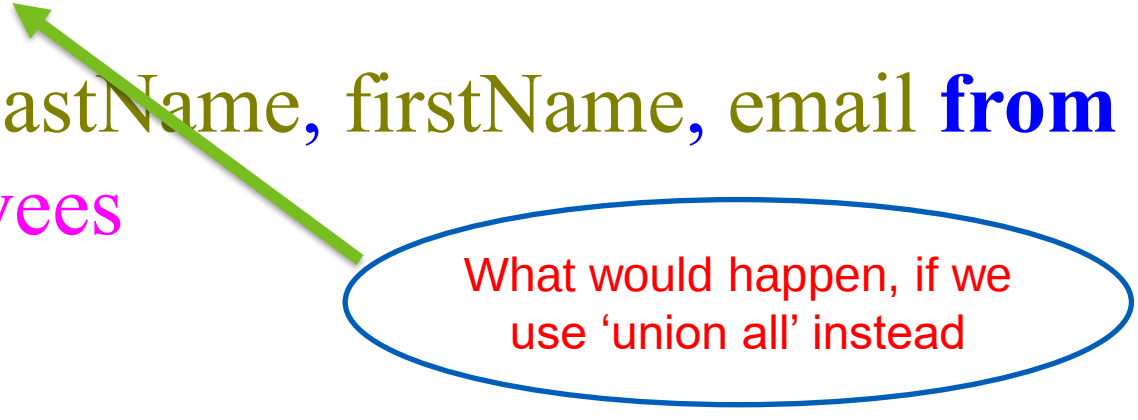
Table employee

Question 2

- Your boss asks you to provide the contact details of both customers and employees in the USA, including their names and contacts (e.g., phone number or email).
- Assuming that your company is located in the USA, all the employees should be on the list. However, customers may not be from the USA.

Solution

```
select contactLastName, contactFirstName,
phone as contact from customers where country
like 'USA'
union
select lastName, firstName, email from
employees
```



Get a fixed value on a select

```
(Select contactLastName, contactFirstName,
phone as contact, "customer" AS category
from customers where country like 'USA' limit 5)

union

select lastName, firstName, email,
"employee" AS category
from employees limit 10
```

productCode	priceEach
S72_3212	43,68
S72_3212	44,23
S72_3212	44,77
S72_3212	46,96
S72_3212	47,5
S72_3212	48,05
S72_3212	48,59
S72_3212	49,14
S72_3212	51,32
S72_3212	51,87
S72_3212	52,42
S72_3212	52,96
S72_3212	53,51
S72_3212	54,05
S72_3212	54,6
S72_1253	39,73
S72_1253	40,22
S72_1253	41,22
S72_1253	41,71
S72_1253	42,71
S72_1253	43,2
S72_1253	43,7
S72_1253	44,2

Select…Group by…

- **"Group by" likes "Distinct" to offer unique record based on columns specified.**
- **select distinct productCode, priceEach from orderdetails**

The same results

- **select productCode, priceEach from orderdetails group by productCode, priceEach**

GROUP BY (Aggregate) Functions

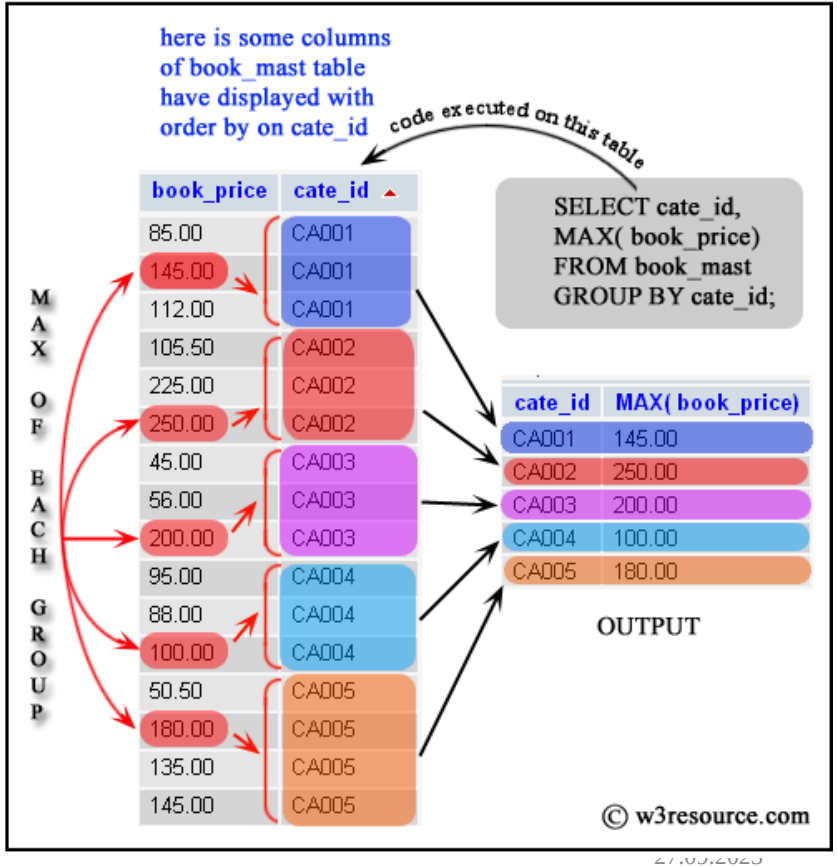
Name	Description
AVG()	Return the average value of the argument
COUNT()	Return a count of the number of rows returned
COUNT(DISTINCT)	Return the count of a number of different values
GROUP_CONCAT()	Return a concatenated string
MAX()	Return the maximum value
MIN()	Return the minimum value
STD()	Return the population standard deviation
SUM()	Return the sum
VARIANCE()	Return the population standard variance

Group by (II)

book_id	book_name	cate_id	book_price
BK001	Introduction to Electrodynamics	CA001	85.00
BK002	Understanding of Steel Construction	CA002	105.50
BK003	Guide to Networking	CA003	200.00
BK004	Transfer of Heat and Mass	CA002	250.00
BK005	Conceptual Physics	CA001	145.00
BK006	Fundamentals of Heat	CA001	112.00
BK007	Advanced 3d Graphics	CA003	56.00
BK008	Human Anatomy	CA005	50.50

Table Name: book_mast

```
SELECT cate_id, MAX(book_price)
FROM book_mast
GROUP BY cate_id;
```



Reference: https://www.w3resource.com/mysql/aggregate-functions-and-grouping/aggregate-functions-and-grouping-group_concat.php

orderNumber	productCode	quantityOrdered	priceEach
10,100	S18_1749	30	136
10,100	S18_2248	50	55.09
10,100	S18_4409	22	75.46
10,100	S24_3969	49	35.29
10,110	S18_1589	37	118.22
10,110	S18_1749	42	153
10,110	S18_2248	32	51.46
10,110	S18_2325	33	115.69
10,110	S18_2795	31	163.69
10,110	S18_4409	28	81.91
10,110	S18_4933	42	62

This table offers the information of products included in each sales order.

Table orderdetails

- In one purchase (or a sales order), a customer may buy several products.
- The same product can be sold in different prices in different sales orders.

Attention! A common mistake!

If we can select the maximum quantity ordered of each product via the following code,…

```
SELECT productCode, MAX(quantityOrdered)
FROM orderdetails
GROUP BY productCode
```

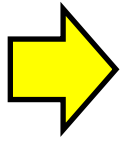
How to retrieve the price of the product that has the largest number in quantity ordered for each order number?

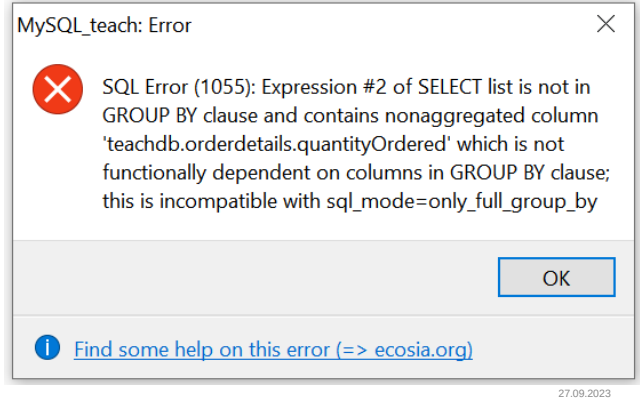
orderNumber	productCode	quantityOrdered	priceEach
10,417	S10_1678	66	79.43
10,251	S10_1678	59	93.79
10,159	S10_1678	49	81.35
10,188	S10_1678	48	95.7
10,318	S10_1678	46	84.22
10,145	S10_1678	45	76.56
10,275	S10_1678	45	81.35
10,329	S10_1678	42	80.39
10,354	S10_1678	42	84.22
10,388	S10_1678	42	80.39

Raw data

… what would be the output for the following queue?

```
SELECT productCode, MAX(quantityOrdered),
priceEach FROM orderdetails GROUP BY
productCode;
```





AVG() + group by

orderNumber	productCode	quantityOrdered	priceEach
10,100	S18_1749	30	136
10,100	S18_2248	50	55.09
10,100	S18_4409	22	75.46
10,100	S24_3969	49	35.29
10,110	S18_1589	37	118.22
10,110	S18_1749	42	153
10,110	S18_2248	32	51.46
10,110	S18_2325	33	115.69
10,110	S18_2795	31	163.69
10,110	S18_4409	28	81.91
10,110	S18_4933	42	62

Revenue generated by a product when ordered =

quantityOrdered X priceEach

What is the average revenue generated by each product when ordered?

Table orderdetails

Improving the presentation of result?

```
SELECT productCode, AVG(priceEach*quantityOrdered) FROM
orderdetails GROUP BY productCode
```

orderdetails (2×109)

productCode	AVG(priceEach*quantityOrdered)
S10_1678	3,219.92035714286
S10_1949	6,786.35571428571
S10_2016	3,928.52928571428
S10_4698	6,095.92857142857
S10_4757	4,568.72571428571
S10_4962	4,397.25035714286
S12_1099	5,982.6474074074
S12_1108	7,065.03185185185
S12_1666	4,253.04464285714
S12_2823	4,848.8225
S12_3148	4,902.36259259259

Query improvement

```
SELECT productCode, AVG(priceEach*quantityOrdered) FROM
orderdetails GROUP BY productCode
```

```
SELECT productCode, ROUND(AVG(priceEach*quantityOrdered), 2)
FROM orderdetails GROUP BY productCode
```

```
SELECT productCode,
ROUND(AVG(priceEach*quantityOrdered), 2) AS avg_revenue
FROM orderdetails
GROUP BY productCode
ORDER BY avg_revenue DESC
```

Count() + group by

orderNumber	productCode	quantityOrdered	priceEach
10 100	S18_1749	30	136
10 100	S18_2248	50	55,09
10 100	S18_4409	22	75,46
10 100	S24_3969	49	35,29
10 101	S18_2325	25	108,06
10 101	S18_2795	26	167,06
10 101	S24_1937	45	32,53
10 101	S24_2022	46	44,35
10 102	S18_1342	39	95,55
10 102	S18_1367	41	43,13
10 103	S10_1949	26	214,3
10 103	S10_4962	42	119,67
10 103	S12_1666	27	121,64
10 103	S18_1097	35	94,5

Table orderdetails

- What is the product that has been most often purchased by customers?

Answer

- **select productCode, count(*) as frequency from orderdetails group by productCode order by frequency desc**

productCode	frequency
S18_3232	53
S24_1578	28
S700_2834	28
S18_2432	28

You can use number to indicate column

```
Select productCode, count(*) as frequency
from orderdetails group by productCode order
by frequency desc
```

… is the same to …

```
Select productCode, count(*) as frequency
from orderdetails group by 1 order by 2 desc
```

Count(Distinct)+ Group by

orderNumber	productCode	quantityOrdered	priceEach
10 100	S18_1749	30	136
10 100	S18_2248	50	55,09
10 100	S18_4409	22	75,46
10 100	S24_3969	49	35,29
10 101	S18_2325	25	108,06
10 101	S18_2795	26	167,06
10 101	S24_1937	45	32,53
10 101	S24_2022	46	44,35
10 102	S18_1342	39	95,55
10 102	S18_1367	41	43,13
10 103	S10_1949	26	214,3
10 103	S10_4962	42	119,67
10 103	S12_1666	27	121,64
10 103	S18_1097	35	94,5

Table orderdetails

A product can be sold in different prices in different sales orders.

- **What is the product that has been sold in the largest amount of different prices?**

Answer

```
select productCode, count(distinct priceEach)
as frequency
from orderdetails
group by productCode
order by frequency desc
```

productCode	frequency
S24_1444	19
S32_1268	19
S18_3232	19
S700_2610	18
S32_2509	18
S10_4757	18
S18_1984	18
S12_1666	17
S18_2319	17
S24_3420	17

group_concat()+ Group by

- **This function returns a string result with the concatenated non-NULL values from a group.**

customerNumber	customerName	contactLastName	contactFirstName	country	city
282	Souveniers And Things Co.	Huxley	Adrian	Australia	Chatswood
398	Tokyo Collectables, Ltd	Shimamura	Akiko	Japan	Minato-ku
237	ANG Resellers	Camino	Alejandra	Spain	Madrid
443	Feuer Online Stores, Inc	Feuer	Alexander	Germany	Leipzig
480	Kremlin Collectables, Co.	Semenov	Alexander	Russia	Saint Petersburg
379	Collectables For Less Inc.	Nelson	Allen	USA	Brickhaven
324	Stylish Desk Decors, Co.	Brown	Ann	UK	London
276	Anna's Decorations, Ltd	O'Hara	Anna	Australia	North Sydney
242	Alpha Cognac	Roulet	Annette	France	Toulouse
356	SAR Distributors, Co	Kuger	Armand	South Africa	Hatfield
385	Cruz & Sons Co.	Cruz	Arnold	Philippines	Makati City
333	Australian Gift Network, Co	Calaghan	Ben	Australia	South Brisbane
303	Schuyler Imports	Schuyler	Bradley	Netherlands	Amsterdam
376	Precious Collectables	Urs	Braun	Switzerland	Bern

An example

- **select group_concat(contactFirstName, contactLastName) from customers**

```
Carine Schmitt,JeanKing,PeterFerguson,Janine Labrune,Jonas Bergulfsen,SusanNelson,Zbyszek
Piestrzeniewicz,RolandKeitel,JulieMurphy,KwaiLee,Diego Freyre,Christina Berglund,Jytte Petersen,Mary
Saveley,EricNatividad,JeffYoung,KelvinLeong,JuriHashimoto,WendyVictorino,VeyselOeztan,KeithFranco,Isabel de Castro,Martine
Rancé,MarieBertrand,JerryTseng,JulieKing,MoryKentary,MichaelFrick,MattiKarttunen,RachelAshworth,DeanCassidy,LeslieTaylor,ElizabethDevon,Yoshi
Tamuri,MiguelBarajas,JulieYoung,BrydeyWalker,Frédérique Citeaux,MikeGao,Eduardo Saavedra,MaryYoung,Horst Kloss,PalleIbsen,Jean
Fresnière,Alejandra Camino,ValarieThompson,Helen Bennett,Annette Roulet,Renate Messner,Paolo Accorti,DanielDa Silva,Daniel Tonini,Henriette
Pfalzheim,Elizabeth Lincoln,Peter Franken,AnnaO'Hara,Giovanni
Rovelli,AdrianHuxley,MartaHernandez,EdHarrison,MihaelHolz,JanKlaeboe,BradleySchuyler,MelAndersen,PirkkoKoskitalo,Catherine
Dewey,SteveFrick,WingHuang,JulieBrown,MikeGraham,Ann Brown,WilliamBrown,BenCalaghan,KalleSuominen,Philip Cramer,Fran
```

customerNumber	customerName	contactLastName	contactFirstName	country	city
282	Souveniers And Things Co.	Huxley	Adrian	Australia	Chatswood
398	Tokyo Collectables, Ltd	Shimamura	Akiko	Japan	Minato-ku
237	ANG Resellers	Camino	Alejandra	Spain	Madrid
443	Feuer Online Stores, Inc	Feuer	Alexander	Germany	Leipzig
480	Kremlin Collectables, Co.	Semenov	Alexander	Russia	Saint Petersburg
379	Collectables For Less Inc.	Nelson	Allen	USA	Brickhaven

Group_concat

France	Marseille	Laurence Lebihan
France	Nantes	Janine Labrune, Carine Schmitt
France	Paris	Daniel Da Silva, Marie Bertrand, Dominique Perrier

```
select country, city,
group_concat(' ', contactFirstName,' ',contactLastName)
as contact_list
from customers group by country, city
```

You find duplicated records mistakenly appearing at the table below. How can you obtain a clean table by making each payment appears only once?

customerNumber	checkNumber	paymentDate	amount
103	HQ336336	2004-10-19	6,066.78
103	JM555205	2003-06-05	14,571.44
103	OM314933	2004-12-18	1,676.14
112	BO864823	2004-12-17	14,191.12
112	HQ55022	2003-06-06	32,641.98
112	ND748579	2004-08-20	33,347.88
114	GG31455	2003-05-20	45,864.03
114	MA765515	2004-12-15	82,261.22
114	NP603840	2003-05-31	7,565.08
114	NR27552	2004-03-10	44,894.74
119	DB933704	2004-11-14	19,501.82
119	LN373447	2004-08-08	47,924.19

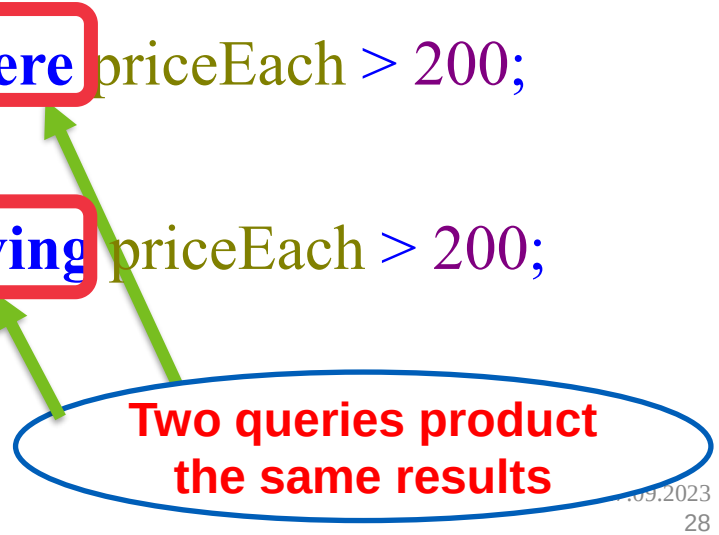
select * from payments **group by** customerNumber, checkNumber, paymentDate, amount

Select…Group by + having

- **Where and Having are similar if Group By is not included in the command.**

```
select priceEach from orderdetails where priceEach > 200;

select priceEach from orderdetails having priceEach > 200;
```



Where versus Having


```
select priceEach as p from orderdetails where p > 200
```

```
select priceEach as p from orderdetails having p > 200;
```

- **WHERE is applied before Select or Group by, while HAVING is applied after.**

Example

```
select orderNumber, count(*) as freq
from orderdetails
where priceEach > 50
group by orderNumber having freq > 10;
```

orderNumber	productCode	quantityOrdered	priceEach
10 100	S18_1749	30	136
10 100	S18_2248	50	55,09
10 100	S18_4409	22	75,46
10 100	S24_3969	49	35,29
10 101	S18_2325	25	108,06
10 101	S18_2795	26	167,06
10 101	S24_1937	45	32,53
10 101	S24_2022	46	44,35
10 102	S18_1342	39	95,55

Order of keyword operation

- **Select command template (a simple version)**

Select (columns or computed new columns)

From (table[s])

Where (conditions)

Group by (columns or computed new columns)

Having (conditions – based on computed new columns, e.g. count)

Order by (columns or computed new columns)

Limit (number)

- **Sequence of operation:**

From → Where → Group by → Select → Having → Order by → Limit

NULL value VS. Empty value

ID	First_Name	Middel_Name	Last_Name
1	Francis	Lee	Bailey
2	James	Lance	Bass
3	Monte		Heilig
4	Minta	(NULL)	Lofton

Manipulation on NULL value

- **Selection of empty value or purely space**

Select * from peoplenames where Middel_Name = '';

ID	First_Name	Middel_Name	Last_Name
3	Monte		Heilig

!= '';

- **Selection of NULL value**

Select * from peoplenames where Middel_Name is null;

ID	First_Name	Middel_Name	Last_Name
4	Minta	(NULL)	Lofton

is not null

Manipulation on date

- Business-oriented data is normally **time- or date-based** by assigning a time stamp to record the occurrence of each event, e.g.:
  - Supermarket receipt
  - Time of delivery or making order
  - Stock exchange
- current_time() function in MySQL.

Possible research questions

- **Is the revenue generated on Monday higher than that on Tuesday?**

- **Did consumers complaint more often in the weekends than weekdays?**

DATE(*expr*)

- **Business data is often as accurate as seconds.**
- **Extracting the DATE part of a datetime expression *expr*.**

```
mysql> SELECT DATE('2003-12-31 01:02:03');
        -> '2003-12-31'
```

extract information

Command	Result
select hour('2015-03-16 23:45:59');	23
select minute('2015-03-16 23:45:59');	45
select second('2015-03-16 23:45:59');	59
select day('2015-03-16 23:45:59');	16
select week('2015-03-16 23:45:59');	11
select month('2015-03-16 23:45:59');	3
select quarter('2015-03-16 23:45:59');	1
select year('2015-03-16 23:45:59');	2015

DAYNAME(*date*)
MONTHNAME(*date*)

- **DAYNAME(*date*);**

```
mysql> SELECT DAYNAME('2007-02-03');
        -> 'Saturday'
```

- **MONTHNAME(*date*)**

```
mysql> SELECT MONTHNAME('2008-02-03');
        -> 'February'
```

Possible research question

- **In a week, when will consumers most likely submit their complaints to CFPB?**
  - **DayName**
  - **Group by**

http://presemo.aalto.fi/drm/

ZIP_code	State	Submitted_via	Data_received	Data_sent_to_company	Company
0		Phone	2011-12-02	2011-12-08	Capital One
0		Referral	2011-12-05	2011-12-05	GE Capital Retail

Column: Data_received Table name: Tablex

Answer

```
select dayname(Data_received),
count(*) as freq
from tablex
group by dayname(Data_received)
order by freq desc
```

weekday() vs. dayofweek()

- **For weekday(): 0 = Monday, 1 = Tuesday, 2 = Wednesday, 3 = Thursday, 4 = Friday, 5 = Saturday, 6 = Sunday.**
- **For dayofweek(): 1=Sunday, 2=Monday, 3=Tuesday, 4=Wednesday, 5=Thursday, 6=Friday, 7=Saturday.**

Examples

- **Select dayofweek**("2017-06-15");
  - Return : 5
- **Select weekday**("2017-06-15");
  - Return : 3
- **Select dayname**("2017-06-15");
  - Return: **Thursday**

Manipulation on date VS. number

- **select** '2008-01-02' **-** 1

'2008-01-02' -1
2 007

- **select** '2008-01-02' **-** '2007-11-01'

'2008-01-02' -'2007-11-01'
1

Normal mathematic calculation like + and - cannot be applied to date directly.

DATE_ADD(*date*, INTERVAL *expr unit*)

- **DATE_ADD() is a synonym for ADDDATE()**

```
mysql> SELECT DATE_ADD('2008-01-02', INTERVAL 31 DAY);
        -> '2008-02-02'
mysql> SELECT ADDDATE('2008-01-02', INTERVAL 31 DAY);
        -> '2008-02-02'
```

Question

- **A person is born on March 16, 1998. When will the date that the person has been living in this world for 10,000 days?**

```
Select date_add('1998-03-16', interval 10000 day)
```

→ 2025-08-01

DATEDIFF(*expr1,expr2*)

- DATEDIFF() returns *expr1 – expr2* expressed as a value in days from one date to the other. *expr1* and *expr2* are date or date-and-time expressions.
- Only the date parts of the values are used in the calculation.

DATEDIFF(*expr1,expr2*)

```
mysql> SELECT DATEDIFF('2007-12-31 23:59:59','2007-12-30');
        -> 1
mysql> SELECT DATEDIFF('2010-11-30 23:59:59','2010-12-31');
        -> -31
```

Question

- **In CFPB, which company has the largest average interval between Date_received and Date_sent_to_company? Only those company who has over 50 records in the data will be considered.**

ZIP_code	State	Submitted_via	Data_received	Data_sent_to_company	Company
0		Phone	2011-12-02	2011-12-08	Capital One
0		Referral	2011-12-05	2011-12-05	GE Capital Retail

Table name: Tablex

Answer

```
SELECT company,
avg(DATEDIFF(Data_sent_to_company, Data_received)) AS diff,
COUNT(*) AS freq
FROM tablex
GROUP BY company
HAVING freq > 50
ORDER BY diff DESC
```

Alter table

- **Alter Table table_name Add column_name datatype**
- **Alter Table table_name Drop column_name**

Update table

- **Update** table_name

  Set column_name1 = value|expression,

  column_name2 = value|expression,

  …

  column_nameN = value|expression

  Where conditions;

productCode	productName	quantityInStock	buyPrice	MSRP
S10_1678	1969 Harley Davidson Ultima...	7933	48.81	95.7
S10_1949	1952 Alpine Renault 1300	7305	98.58	214.3
S10_2016	1996 Moto Guzzi 1100i	6625	68.99	118.94
S10_4698	2003 Harley-Davidson Eagle ...	5582	91.02	193.66
S10_4757	1972 Alfa Romeo GTA	3252	85.68	136
S10_4962	1962 LanciaA Delta 16V	6791	103.42	147.74

Table products

- **Price_difference = (MSRP-buyPrice)**

Please create a new column of Price_difference

```
Alter table products add Price_difference decimal(10,2);

Update products set Price_difference = (MSRP-buyPrice);
```

```
Update products set Price_difference = (MSRP-buyPrice)
Where productName like '1996 Moto Guzzi%';
```

Tips

- **Update versus Select**
  - **‘Select’ is just a presentation of new result while ‘update’ actually changes and saves the data.**

- **Undo the last action?! It does not work!**

Delete records from table

- **Delete from table_name**

  [where conditions]

- **Delete from table_name**
  - **This commands will remove all the records from the table [output: an empty table]**

Example

- **In table customers, some values of the ‘salesRepEmplooyeeNumber’ column are null**

```
select * from customers
where salesRepEmployeeNumber is null

Delete from customers
where salesRepEmployeeNumber is null
```

Sub-Queries (1)

- **Template 1:**

```
Create table TB_name as
(Select attributes
        from table or view
        [Where  conditions]
        [Group by attributes [Having condition]]
        [Order by attributes [asc | desc]]
        [Limit])
```

Sub-Queries (2)

- If the result of the **select** command is based on **one column of another table.** E.g.:

```
Select attributes
        from table_1
        Where attributes IN| NOT IN
        (Select ONE_column
                from table_2
                Where attributes )
```

Example

- **Please provide the contact information of customers who made a payment over 100,000 Euro.**

customerNumber	country	creditLimit	contactLastName	contactFirstName
103	France	21,000	Schmitt	Carine
112	USA	71,800	King	Jean
114	Australia	117,300	Ferguson	Peter
119	France	118,200	Labrune	Janine
121	Norway	81,700	Bergulfsen	Jonas
124	USA	210,500	Nelson	Susan
125	Poland	0	Piestrzeniewicz	Zbyszek
128	Germany	59,700	Keitel	Roland

customers

customerNumber	checkNumber	paymentDate	amount
363	IS232033	2003-01-16	10 223,83
128	DI925118	2003-01-28	10 549,01
181	GQ132144	2003-01-30	5 494,78
121	DB889831	2003-02-16	50 218,95
145	JJ246391	2003-02-20	53 959,21
141	JN722010	2003-02-25	40 206,2
278	GP636783	2003-03-02	52 151,81
385	EK785462	2003-03-09	51 001,22

payments

Example

```
Select *
    from customers
    where customerNumber in
        (select customerNumber
            from payments
            where amount > 100000)
```

Question

- **Retrieve the payment information of the customers who are living in the country, Spain, with a creditLimit of over 5000?**

customerNumber	country	creditLimit	contactLastName	contactFirstName
103	France	21,000	Schmitt	Carine
112	USA	71,800	King	Jean
114	Australia	117,300	Ferguson	Peter
119	France	118,200	Labrune	Janine
121	Norway	81,700	Bergulfsen	Jonas
124	USA	210,500	Nelson	Susan
125	Poland	0	Piestrzeniewicz	Zbyszek
128	Germany	59,700	Keitel	Roland

customers

customerNumber	checkNumber	paymentDate	amount
363	IS232033	2003-01-16	10 223,83
128	DI925118	2003-01-28	10 549,01
181	GQ132144	2003-01-30	5 494,78
121	DB889831	2003-02-16	50 218,95
145	JJ246391	2003-02-20	53 959,21
141	JN722010	2003-02-25	40 206,2
278	GP636783	2003-03-02	52 151,81
385	EK785462	2003-03-09	51 001,22

payments

Answer

```
SELECT *
FROM payments
WHERE customerNumber IN (

SELECT customerNumber
FROM customers
WHERE country = 'Spain' AND
creditLimit > 5000
)
```

Sub-Queries (2.1)

- If the result of the **select** command is based on **multiple columns of another table.** E.g.:

```
Select attributes
        from table_1
        Where (attribute1, … , attributeN) IN| NOT IN
        (Select column1, … , columnN
                from table_2
                Where attributes )
```

Question

- Assume we have a table of undelivered products, how can we calculate the revenue of those delivered products.

orderNumber	productCode
10 107	S10_1678
10 121	S10_1678
10 134	S10_1678
10 145	S10_1678
10 159	S10_1678
10 168	S10_1678
10 180	S10_1678
10 188	S10_1678
10 201	S10_1678
10 211	S10_1678
10 223	S10_1678
10 237	S10_1678
10 251	S10_1678
10 263	S10_1678

Table: undelivered_products

orderNumber	productCode	quantityOrdered	priceEach
10 100	S18_1749	30	136
10 100	S18_2248	50	55,09
10 100	S18_4409	22	75,46
10 100	S24_3969	49	35,29
10 101	S18_2325	25	108,06
10 101	S18_2795	26	167,06
10 101	S24_1937	45	32,53
10 101	S24_2022	46	44,35
10 102	S18_1342	39	95,55
10 102	S18_1367	41	43,13
10 103	S10_1949	26	214,3
10 103	S10_4962	42	119,67
10 103	S12_1666	27	121,64
10 103	S18_1097	35	94,5
10 103	S18_2432	22	58,34
10 103	S18_2949	27	92,19
10 103	S18_2957	35	61,84
10 103	S18_3136	25	86,92

Table: orderdetails

http://stackoverflow.com/questions/8435107/mysql-where-not-in-using-two-columns

orderNumber	productCode
10 107	S10_1678
10 121	S10_1678
10 134	S10_1678
10 145	S10_1678
10 159	S10_1678
10 168	S10_1678
10 180	S10_1678
10 188	S10_1678
10 201	S10_1678
10 211	S10_1678
10 223	S10_1678
10 237	S10_1678

Table: undelivered_products

orderNumber	productCode	quantityOrdered	priceEach
10 100	S18_1749	30	136
10 100	S18_2248	50	55,09
10 100	S18_4409	22	75,46
10 100	S24_3969	49	35,29
10 101	S18_2325	25	108,06
10 101	S18_2795	26	167,06
10 101	S24_1937	45	32,53
10 101	S24_2022	46	44,35
10 102	S18_1342	39	95,55
10 102	S18_1367	41	43,13
10 103	S10_1949	26	214,3
10 103	S10_4962	42	119,67

Table: orderdetails

```
Select sum(quantityOrdered*priceEach)
FROM orderdetails
WHERE (orderNumber,productCode) NOT IN
(select orderNumber,productCode FROM
undelivered_products)
```

Attention! A common mistake!

If we can select the maximum quantity ordered of each product via the following code,…

```
SELECT productCode, MAX(quantityOrdered)
FROM orderdetails
GROUP BY productCode
```

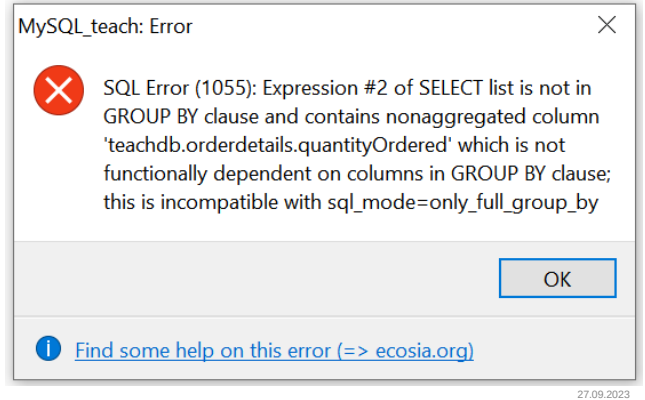
How to retrieve the price of the product that has the largest number in quantity ordered for each order number?

orderNumber	productCode	quantityOrdered	priceEach
10,417	S10_1678	66	79.43
10,251	S10_1678	59	93.79
10,159	S10_1678	49	81.35
10,188	S10_1678	48	95.7
10,318	S10_1678	46	84.22
10,145	S10_1678	45	76.56
10,275	S10_1678	45	81.35
10,329	S10_1678	42	80.39
10,354	S10_1678	42	84.22
10,388	S10_1678	42	80.39

Raw data

… what would be the output for the following queue?

```
SELECT productCode, MAX(quantityOrdered),
priceEach FROM orderdetails GROUP BY
productCode;
```



Answer to the question for common mistake of “group by”

The command to extract rows with the ‘priceEach’ of each productCode with maximam quantityOrdered

```
Select * from orderdetails
WHERE (productCode,quantityOrdered) IN
(SELECT productCode, MAX(quantityOrdered)
FROM orderdetails GROUP BY productCode);
```

Think and solution

How to select the records with the second largest quantityOrdered for each productCode?

orderNumber	productCode	quantityOrdered	priceEach
10,417	S10_1678	66	79.43
10,251	S10_1678	59	93.79
10,159	S10_1678	49	81.35
10,188	S10_1678	48	95.7
10,318	S10_1678	46	84.22
10,145	S10_1678	45	76.56
10,275	S10_1678	45	81.35
10,329	S10_1678	42	80.39
10,354	S10_1678	42	84.22
10,388	S10_1678	42	80.39

Solution: if the largest quantityOrdered for each productCode is removed from the table [or use NOT IN], the second largest one becomes the largest one.

Change date format

- **select date_format**('2015-03-16', '%m.%d.%y');

Result: 03.16.15

- **select date_format**('2015-03-16', '%m-%d-%y');

Result: 03-16-15

- **select date_format**('2015-03-16', '%y-%m-%d');

Result: 15-03-16

- **select date_format**('2015-03-16', '%Y-%M-%D');

Result: 2015-March-16th

STR_TO_DATE()

- **This is the inverse of the DATE_FORMAT() function. It takes a string *str* and a format string *format*.**

```
mysql> SELECT STR_TO_DATE('01,5,2013','%d,%m,%Y');
        -> '2013-05-01'
mysql> SELECT STR_TO_DATE('May 1, 2013','%M %d,%Y');
        -> '2013-05-01'
```