SQL – Basics (More Practice)

CS 4750
Database Systems

Example 1: Translate and Clean Up

Consider sample data of an Orders table. How many days elapsed between the order date and the ship date for each order?

Orders

OrderNumber	OrderDate	ShipDate	CustomerID	EmployeeID
1	2017-09-02	2017-09-05	1018	707
2	2017-09-02	2017-09-04	1001	703
3	2017-09-02	2017-09-05	1002	707
4	2017-09-02	2017-09-04	1009	703
5	2017-09-02	2017-09-02	1024	708
6	2017-09-02	2017-09-06	1014	702
7	2017-09-02	2017-09-05	1001	708
8	2017-09-02	2017-09-02	1003	703
9	2017-09-02	2017-09-05	1007	708
10	2017-09-02	2017-09-05	1012	701
11	2017-09-03	2017-09-05	1020	706
12	2017-09-03	2017-09-06	1024	706
13	2017-09-03	2017-09-03	1024	704
14	2017-09-03	2017-09-04	1013	704
15	2017-09-03	2017-09-07	1004	701

```
Select the order number,
order date, ship date, ship
date minus order as
DaysElapsed from the Orders
table
```

```
Select ~~the~~ order number,
order date, ship date, ship
date ~~minus~~ – order as
DaysElapsed from ~~the~~ Orders
~~table~~
```

Example 1: Translate and Clean Up

```
Select the order number,
order date, ship date, ship
date minus – order as
DaysElapsed from the Orders
table
```

```
SELECT OrderNumber,
       OrderDate, ShipDate,
       (ShipDate - OrderDate)
       AS DaysElapsed
FROM Orders;
```

OrderNumber	OrderDate	ShipDate	DaysElapsed
1	2017-09-02	2017-09-05	3
2	2017-09-02	2017-09-04	2
3	2017-09-02	2017-09-05	3
4	2017-09-02	2017-09-04	2
5	2017-09-02	2017-09-02	0
6	2017-09-02	2017-09-06	4
7	2017-09-02	2017-09-05	3
8	2017-09-02	2017-09-02	0
9	2017-09-02	2017-09-05	3
10	2017-09-02	2017-09-05	3
11	2017-09-03	2017-09-05	2

Note: rename the column header

Example 2: Translate and Clean Up

Consider sample data of an Orders table. Show me a list of orders made by each customer in descending date order

Orders

OrderNumber	OrderDate	ShipDate	CustomerID	EmployeeID
1	2017-09-02	2017-09-05	1018	707
2	2017-09-02	2017-09-04	1001	703
3	2017-09-02	2017-09-05	1002	707
4	2017-09-02	2017-09-04	1009	703
5	2017-09-02	2017-09-02	1024	708
6	2017-09-02	2017-09-06	1014	702
7	2017-09-02	2017-09-05	1001	708
8	2017-09-02	2017-09-02	1003	703
9	2017-09-02	2017-09-05	1007	708
10	2017-09-02	2017-09-05	1012	701
11	2017-09-03	2017-09-05	1020	706
12	2017-09-03	2017-09-06	1024	706
13	2017-09-03	2017-09-03	1024	704
14	2017-09-03	2017-09-04	1013	704
15	2017-09-03	2017-09-07	1004	701

Select the customer ID, order number, order date, ship date, from the Orders table for each customer and then sort by customer and descending order date

Select ~~the~~ customer ID, order number, order date, ship date, from ~~the~~ Orders ~~table~~ ~~for each customer~~ group by customer ID ~~and then sort by customer~~ order by customer ID ~~and descending~~ order date desc

Example 2: Translate and Clean Up

Select ~~the~~ customer ID, order number, order date, ship date, from ~~the~~ Orders ~~table~~ ~~for each customer~~ group by customer ID ~~and then sort by customer~~ order by customer ID ~~and descending~~ order date desc

```
SELECT CustomerID,
       OrderNumber, OrderDate,
       ShipDate
FROM Orders
GROUP BY CustomerID,
         OrderNumber
ORDER BY CustomerID,
         OrderDate DESC;
```

CustomerID 1	OrderNumber	OrderDate 2	ShipDate
1001	16	2017-09-03	2017-09-07
1001	2	2017-09-02	2017-09-04
1001	7	2017-09-02	2017-09-05
1002	707	2018-01-18	2018-01-19
1002	693	2018-01-16	2018-01-19
1002	696	2018-01-16	2018-01-17
1002	688	2018-01-15	2018-01-19
1002	676	2018-01-13	2018-01-16
1002	636	2018-01-08	2018-01-11
1002	635	2018-01-08	2018-01-11
1002	634	2018-01-08	2018-01-12
1003	764	2018-01-30	2018-01-30
1003	736	2018-01-25	2018-01-29
1003	638	2018-01-08	2018-01-10
1003	588	2017-12-30	2017-12-30

Note: Several DBMS requires that everything in "ORDER BY" must be in "GROUP BY" and everything in "GROUP BY" must be in "SELECT." (MySQL doesn't seem to enforce these requirements)

Recap 1: SELECT .. FROM .. WHERE

Student_lecture

S_id	Address	Course	Teaching_assistant
1234	57 Hockanum Blvd	Database Systems	Minnie
2345	1400 E. Bellows	Database Systems	Humpty
3456	900 S. Detroit	Cloud Computing	Dumpty
1234	57 Hockanum Blvd	Web Programming Lang.	Mickey
5678	2131 Forest Lake Ln.	Software Analysis	Minnie

Find Courses that Minnie is a TA. Also, list all S_id who takes those courses

3 SELECT
output selected attributes

1 FROM
Open an iterator

```
SELECT S_id, Course FROM Student_lecture AS S
WHERE S.Teaching_assistant = “Minnie” ;
```

2 WHERE
Filter each row

```
For each row in S:
   if (row.Teaching_assistant = ”Minnie”:
      output (row.S_id, row.Course)
```

Recap 2: SELECT .. FROM .. WHERE

Student_lecture

S_id	Address	Course	Teaching_assistant
1234	57 Hockanum Blvd	Database Systems	Minnie
2345	1400 E. Bellows	Database Systems	Humpty
3456	900 S. Detroit	Cloud Computing	Dumpty
1234	57 Hockanum Blvd	Web Programming Lang.	Mickey
5678	2131 Forest Lake Ln.	Software Analysis	Minnie

Find all S_id who is taking Database Systems and have Humpty as a TA

```
SELECT S_id FROM Student_lecture AS S
WHERE  S.Course = “Database Systems” AND
       S.Teaching_assistant = “Humpty” ;
```

S_id
2345

Recap 3: SELECT .. FROM .. WHERE

hiring

TA_id	Name	Year	Two_week_hours
1234	Minnie	4	20
2345	Humpty	3	24
3456	Dumpty	4	30
3333	Minnie	3	12
5678	Mickey	2	16

List all names of TAs and the number of hours the TAs work per week, rename the hours as "Hours_per_week"

```
SELECT Name, two_week_hours/2 AS Hours_per_week
FROM hiring
```

Name	Hours_per_week
Minnie	10
Humpty	12
Dumpty	15
Minnie	6
Mickey	8

Recap 4: SELECT .. FROM .. WHERE

hiring

TA_id	Name	Year	Two_week_hours
1234	Minnie	4	20
2345	Humpty	3	24
3456	Dumpty	4	30
3333	Minnie	3	12
5678	Mickey	2	16

List all years

```
SELECT Year
FROM hiring
```

Year
4
3
4
3
2

Duplicates may occur in output of an operator

Recap 5: SELECT .. FROM .. WHERE

hiring

TA_id	Name	Year	Two_week_hours
1234	Minnie	4	20
2345	Humpty	3	24
3456	Dumpty	4	30
3333	Minnie	3	12
5678	Mickey	2	16

List all years the TAs are in. If multiple TAs are in the same year, list the year only once

```
SELECT DISTINCT Year
FROM hiring
```

Year
4
3
2

Recap 6: SELECT .. FROM .. WHERE

hiring

TA_id	Name	Year	Two_week_hours
1234	Minnie	4	20
2345	Humpty	3	24
3456	Dumpty	4	30
3333	Minnie	3	12
5678	Mickey	2	16

List all names of TAs and the number of hours the TAs work per week, rename the hours as "Hours_per_week", then order the result set by names and then Hours_per_week

```
SELECT Name, two_week_hours/2 AS Hours_per_week
FROM hiring
ORDER BY Name, Hours_per_week
```

Name	Hours_per_week
Dumpty	15
Humpty	12
Mickey	8
Minnie	6
Minnie	10