SKILL PILLS

Skill Pill: Database Manipulation

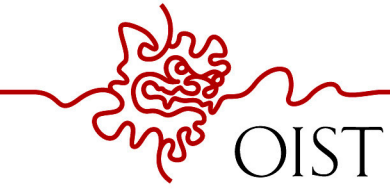
Session 2 – SQL requests (SELECT)

OIST

Introduction

SKILLPILLS

- One of our most common tasks when using a database is getting the data out!
- We use "Structured Query Language" (SQL) to do this
- SQL is a "domain-specific language" for interacting with databases
- MySQL is a database management system (DBMS)
- There are many many (R)DBMSs



Today's Playground

SKILLPILLS

- Today we won't create a database, or any data
- We'll use an existing database hosted on a remote server
- To access the database, use the following commands:

```
$ mysql -u skillpill_user -p -h 10.155.8.10
(enter the password when prompted)
mysql> use skill_pill

mysql> show tables;
(this will list the available Tables)
```

Connection Issues and Explanation

SKILLPILLS

```
$ mysql -u skillpill_user -p -h 10.155.8.10
(enter the password when prompted)
mysql> use skill_pill
```

- 10.155.8.10 is the IP address of my MySQL Server installation
- "-p" instructs mysql to prompt for a password
- If you receive a message like

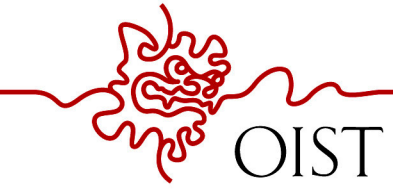
  Command 'mysql' not found, but can be installed with:

  sudo apt install mysql-client-core-5.7

  sudo apt install mariadb-client-core-10.1

  Then you can use one of these commands to install a MySQL client (you don't need to run both)

- "use skill_pill" selects a database within the server. A trailing ";" is optional (for most commands later it will be required!)

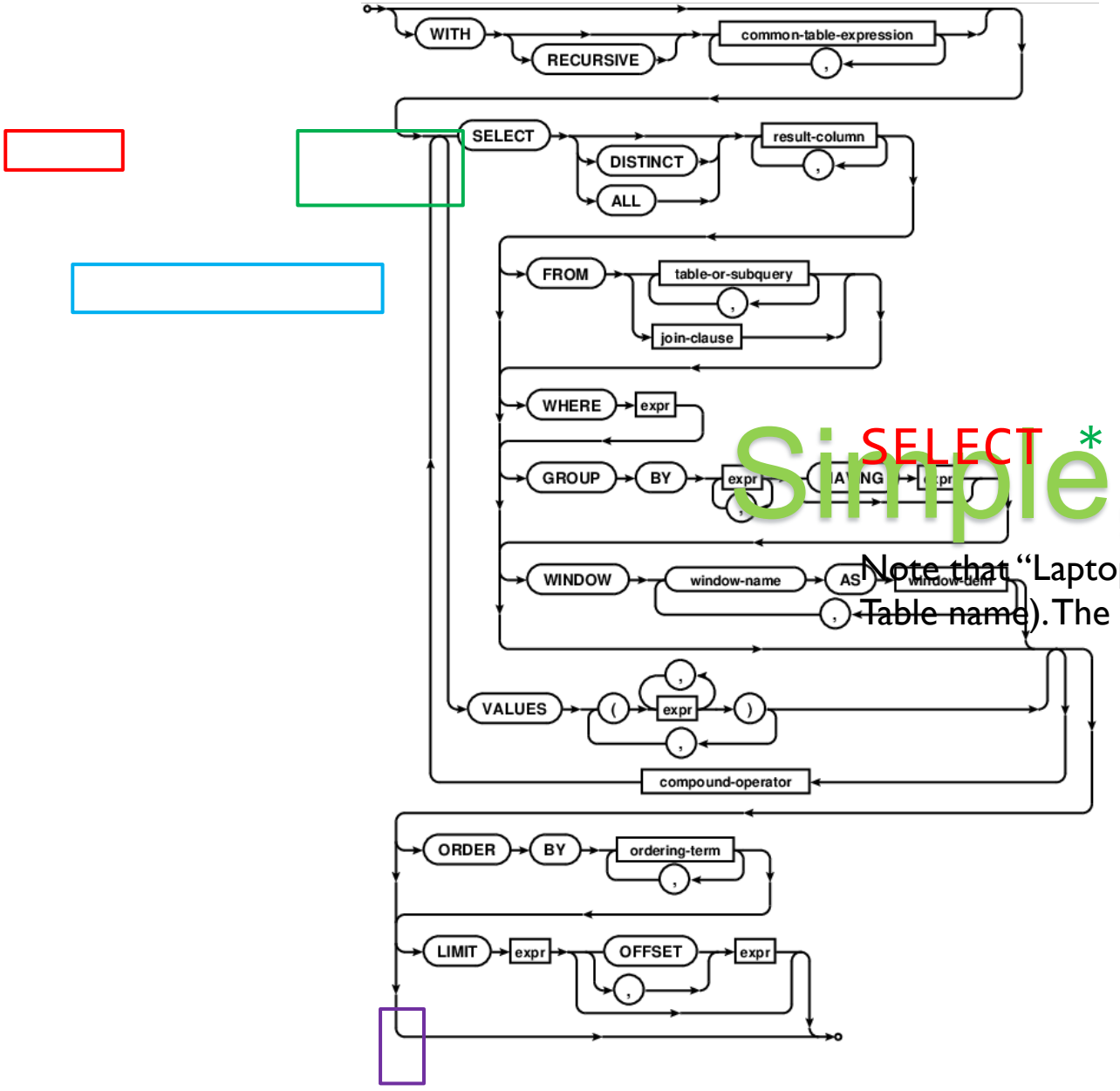


Simple SQL

SKILLPILLS

- SQL has many "keywords" and language elements
- Today we will focus on just one, "SELECT"
- This presentation will use block capitals for SQL keywords
- A nice set of documentation (I think) can be found at https://www.sqlite.org/lang.html

The SELECT Statement



SELECT * FROM Laptop;

Simple, right?

Note that "Laptop" is case sensitive (it is the Table name). The other parts are not.

SELECT * FROM Laptop;

See how I used slightly different capitalization below…

```
mysql> select * from Laptop;
+------+-------+-------+-----+----+---------+--------+
| code | model | speed | ram | hd | price   | screen |
+------+-------+-------+-----+----+---------+--------+
|    1 | 1298  |   350 |  32 |  4 |  700.00 |     11 |
|    2 | 1321  |   500 |  64 |  8 |  970.00 |     12 |
|    3 | 1750  |   750 | 128 | 12 | 1200.00 |     14 |
|    4 | 1298  |   600 |  64 | 10 | 1050.00 |     15 |
|    5 | 1752  |   750 | 128 | 10 | 1150.00 |     14 |
|    6 | 1298  |   450 |  64 | 10 |  950.00 |     12 |
+------+-------+-------+-----+----+---------+--------+
6 rows in set (0.00 sec)
```

But Laptop must have the correct ‘spelling’

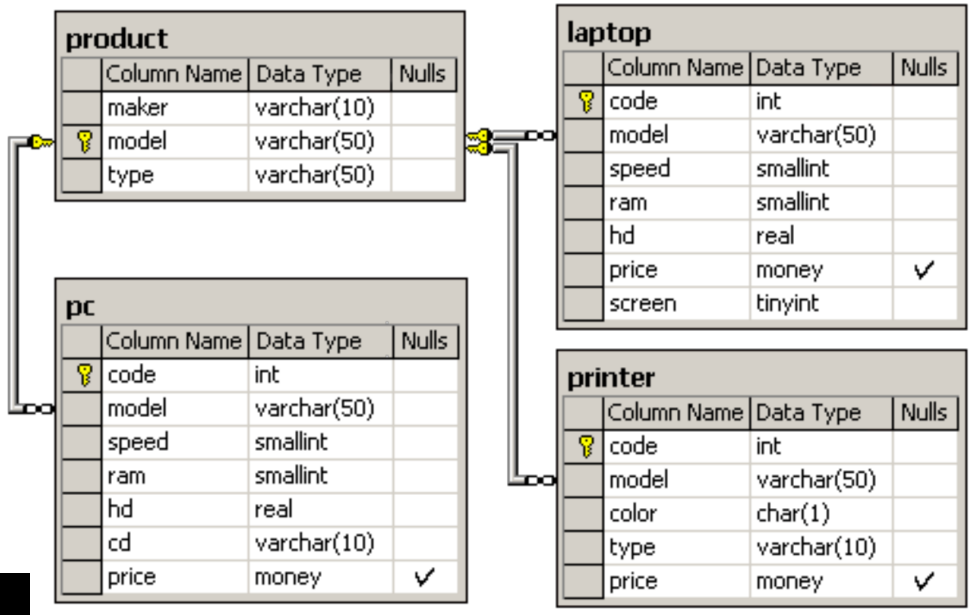
```
mysql> select * from laptop;
ERROR 1146 (42S02): Table 'skill_pill.laptop' doesn't exist
```

- Here's a picture giving an overview of our Database

We can get information like this using the DESCRIBE statement.

For example,
`DESCRIBE PC;`
returns this:

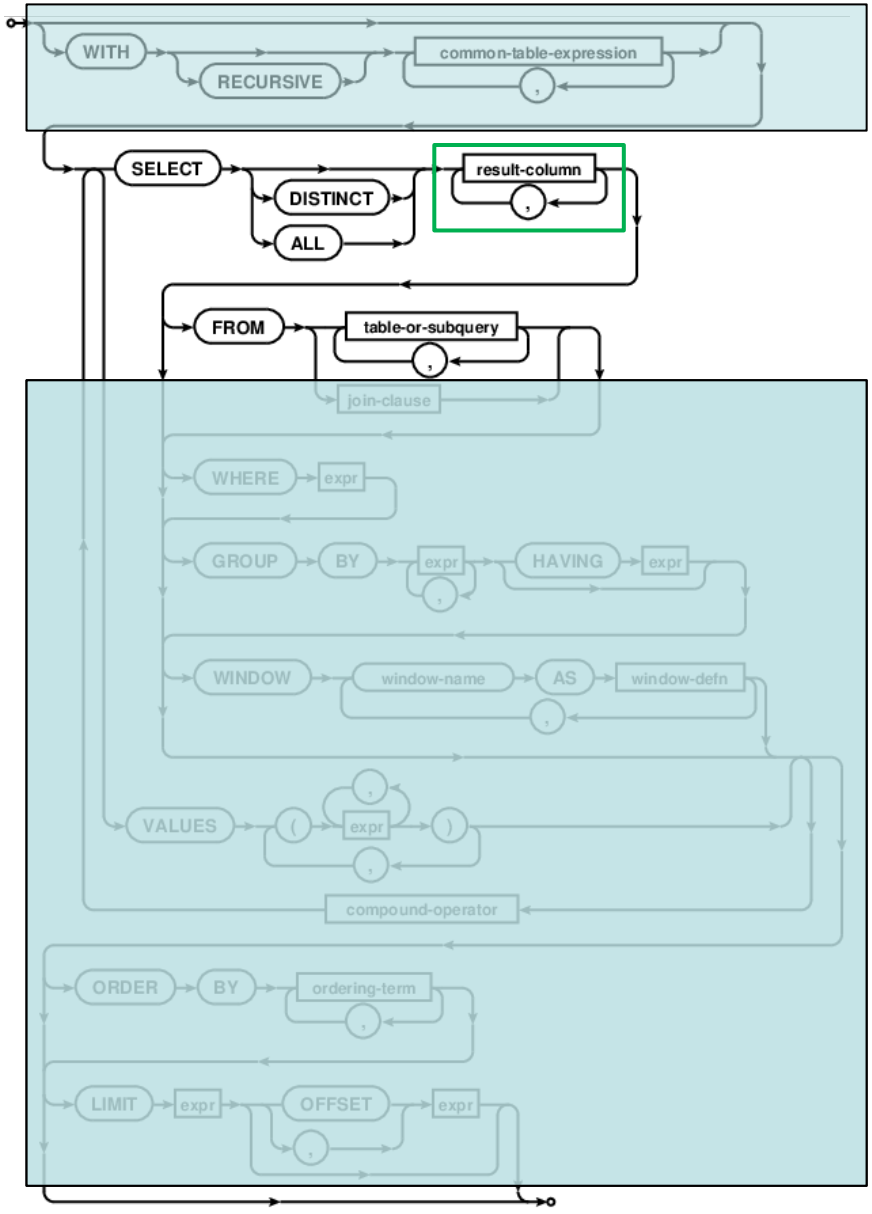
```
mysql> DESCRIBE PC;
+-------+---------------+------+-----+---------+-------+
| Field | Type          | Null | Key | Default | Extra |
+-------+---------------+------+-----+---------+-------+
| code  | int(11)       | NO   |     | NULL    |       |
| model | varchar(50)   | NO   |     | NULL    |       |
| speed | smallint(6)   | NO   |     | NULL    |       |
| ram   | smallint(6)   | NO   |     | NULL    |       |
| hd    | double        | NO   |     | NULL    |       |
| cd    | varchar(10)   | NO   |     | NULL    |       |
| price | decimal(12,2) | YES  |     | NULL    |       |
+-------+---------------+------+-----+---------+-------+
7 rows in set (0.01 sec)
```



http://www.sql-ex.ru

We can also use "`desc PC;`"

The SELECT Statement



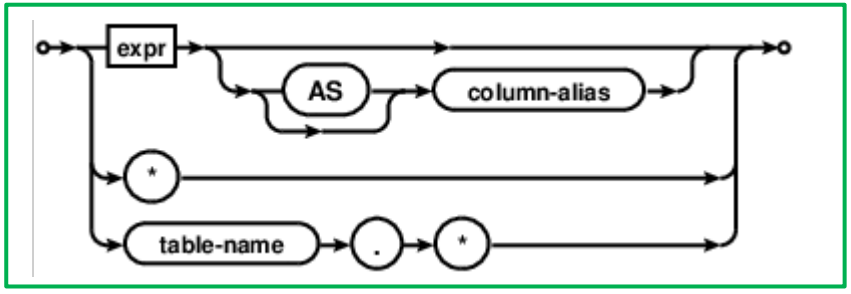
If we don't need all of the columns/fields, we can specify which we want:

```
SELECT model, speed FROM Laptop;
```

Or for more flexibility in display…

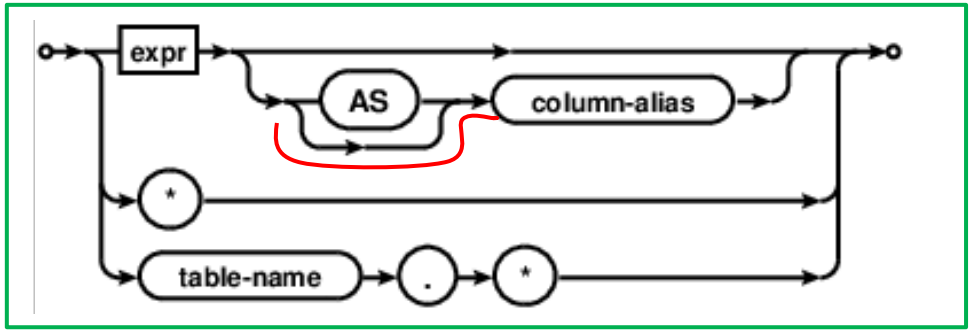
```
SELECT model, speed AS CPU_MHz
FROM Laptop;
```

Here we used "AS" to rename one of the fields, just for the output (the data was unchanged!)



The SELECT Statement

SKILLPILLS



- If we look closely, we can see "AS" is actually optional in this statement…
- Even though you *could* omit it, I suggest using it to make your SQL more readable!

```
mysql> SELECT model, speed as CPU_MHz from Laptop;
+-------+---------+
| model | CPU_MHz |
+-------+---------+
| 1298  |     350 |
| 1321  |     500 |
| 1750  |     750 |
| 1298  |     600 |
| 1752  |     750 |
| 1298  |     450 |
+-------+---------+
6 rows in set (0.00 sec)

mysql> SELECT model, speed CPU_MHz from Laptop;
+-------+---------+
| model | CPU_MHz |
+-------+---------+
| 1298  |     350 |
| 1321  |     500 |
| 1750  |     750 |
| 1298  |     600 |
| 1752  |     750 |
| 1298  |     450 |
+-------+---------+
6 rows in set (0.00 sec)
```

Any fool can write code that a computer can understand.
Good programmers write code that humans can understand.

Martin Fowler (he's a programmer…)

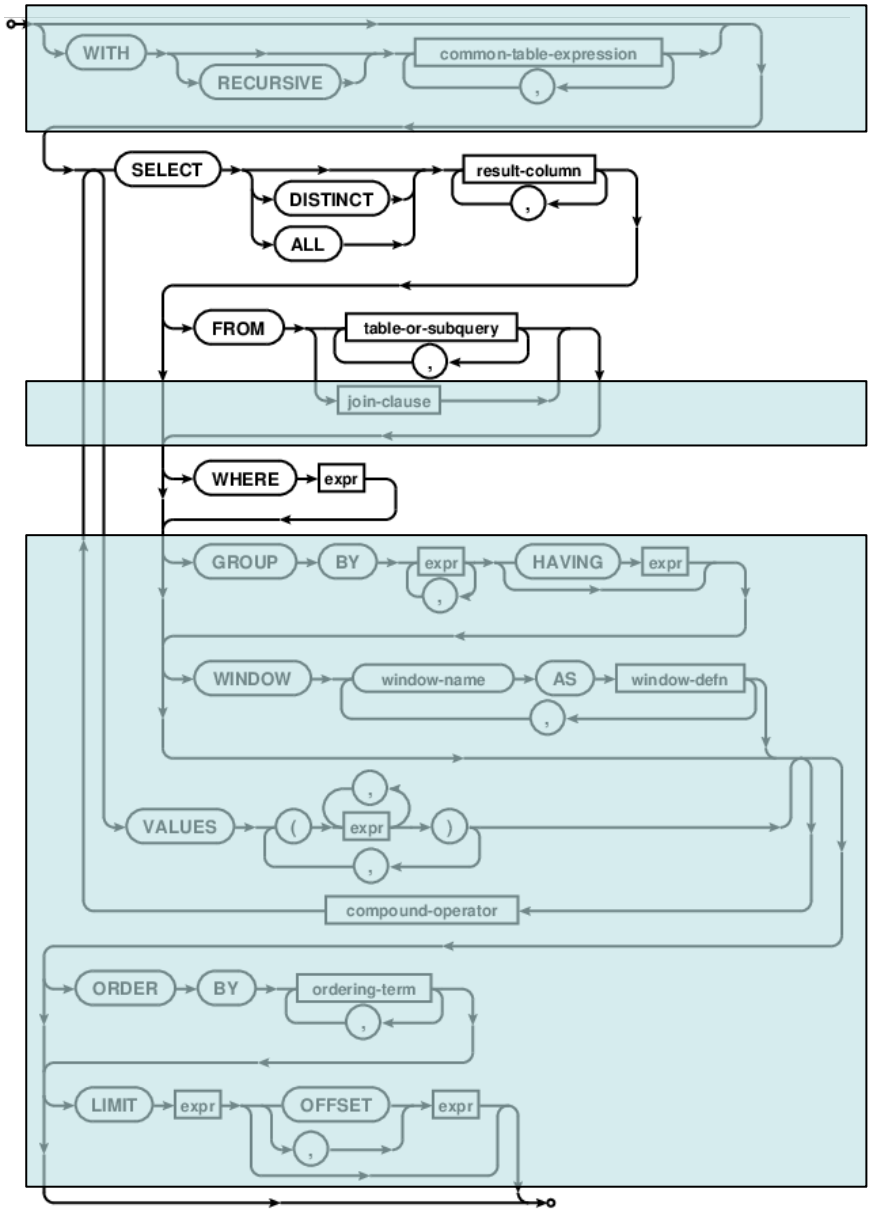
Limiting the Results

SKILLPILLS

- So far we've received all of the 'Records' in a table
- Often we only want a subset that match some criteria

- To do this, we use "WHERE"

```
SELECT * FROM Laptop WHERE
speed > 500;
```



Limiting the Results

SKILLPILLS

- So far we've received all of the 'Records' in a table
- Often we only want a subset that match some criteria

- To do this, we use "WHERE"

```
SELECT * FROM Laptop WHERE speed > 500;
```

- Many comparison statements are possible
  - `>, <, >=, <=, =, !=` (Greater than, less than, …, Equal, Not Equal)
  - `LIKE` (For string comparisons, % is a wildcard character)
  - `NOT` can be used directly to invert the result, e.g.

    ```
    SELECT * FROM Printer WHERE type NOT LIKE 'J%';
    ```
  - `IS NULL, IS NOT NULL`
  - `AND, OR`
  - `IN (value1, value2), NOT IN (…), BETWEEN value1 AND value2`
  - `TRUE, FALSE` (sometimes you can find useful tricks with these… but remember to keep it readable!)

Exercises

SKILLPILLS

1. Find model numbers and speeds for PCs with prices below $500 (assume `price` is in dollars).

   Note you don't see the price if you don't add that to your 'result-columns'

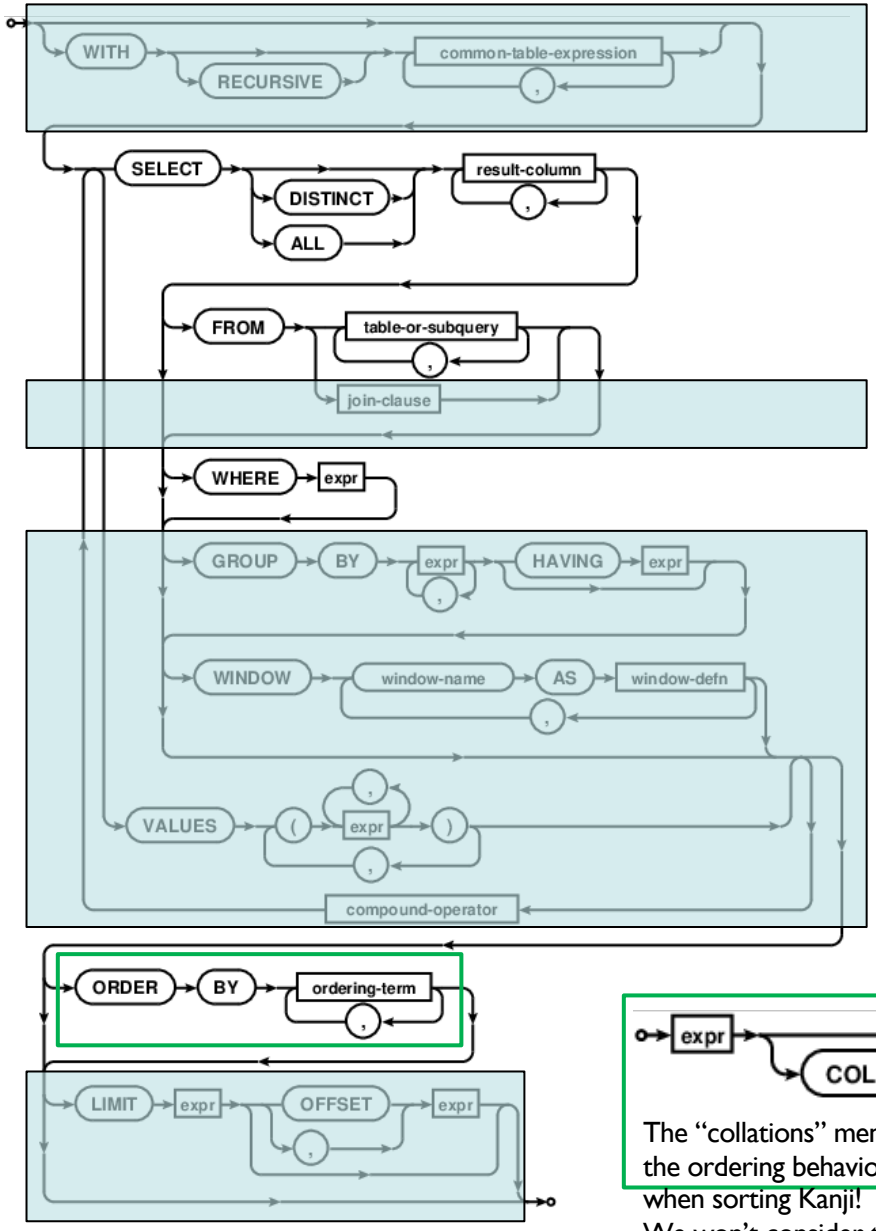
2. List all printer types (hint: try using '`DISTINCT`' immediately after '`SELECT`'

3. Find the model number, speed and hard drive capacity of PCs costing $400 or more, with either a 12x or 24x CD drive

   (Optionally also find PCs costing $400+ that do not have a 12x or 24x CD drive)

4. Find all information about PCs with a CD drive speed multiple of 10x (so 10x, 20x, 30x…)

The ORDER BY Statement



Sometimes (often?) we want the results in a specific order

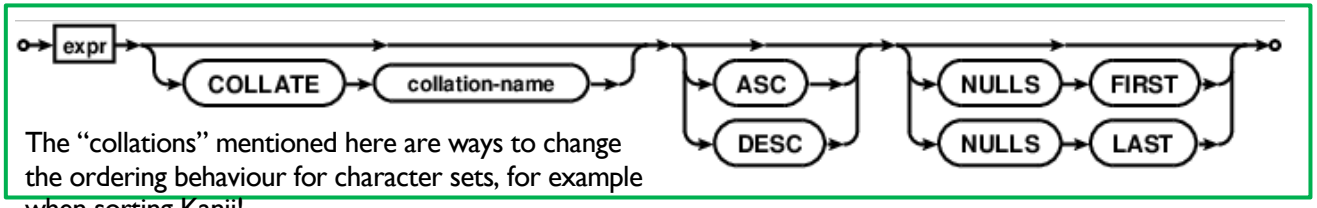
- Biggest first
- Newest first
- Alphabetical

To do this we use "ORDER BY"

```
SELECT * from Laptop ORDER BY speed;
```

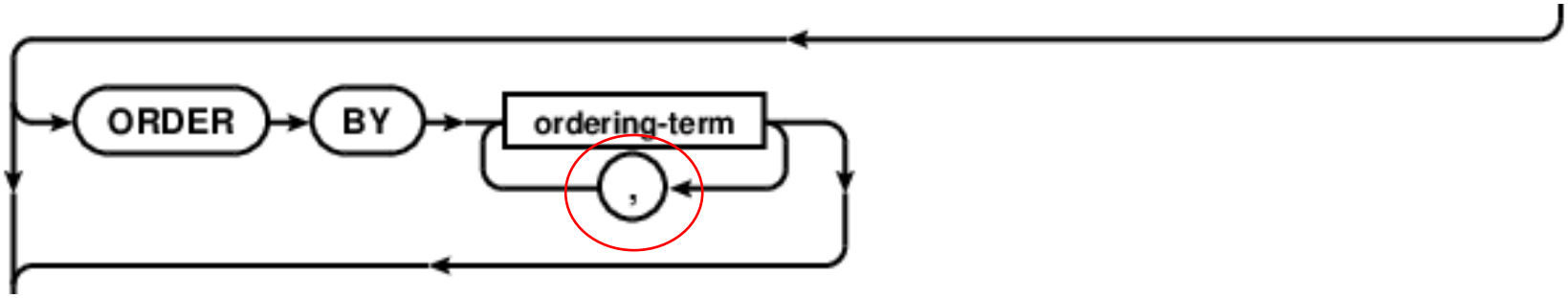
We can specify a direction too (ASC is default):

```
SELECT * from Laptop ORDER BY speed
DESC;
```



The "collations" mentioned here are ways to change the ordering behaviour for character sets, for example when sorting Kanji!
We won't consider them at all…

Multiple ORDER BY elements



- We can see here that multiple "ordering-term"s are allowed
- This lets us break ties in ordering

```
SELECT * from Laptop
ORDER BY speed DESC, hd;
```

```
mysql> select * from Laptop order by speed DESC ;
+------+-------+-------+-----+----+---------+--------+
| code | model | speed | ram | hd | price   | screen |
+------+-------+-------+-----+----+---------+--------+
|    3 | 1750  |   750 | 128 | 12 | 1200.00 |     14 |
|    5 | 1752  |   750 | 128 | 10 | 1150.00 |     14 |
|    4 | 1298  |   600 |  64 | 10 | 1050.00 |     15 |
|    2 | 1321  |   500 |  64 |  8 |  970.00 |     12 |
|    6 | 1298  |   450 |  64 | 10 |  950.00 |     12 |
|    1 | 1298  |   350 |  32 |  4 |  700.00 |     11 |
+------+-------+-------+-----+----+---------+--------+
6 rows in set (0.00 sec)

mysql> select * from Laptop order by speed DESC, hd ;
+------+-------+-------+-----+----+---------+--------+
| code | model | speed | ram | hd | price   | screen |
+------+-------+-------+-----+----+---------+--------+
|    5 | 1752  |   750 | 128 | 10 | 1150.00 |     14 |
|    3 | 1750  |   750 | 128 | 12 | 1200.00 |     14 |
|    4 | 1298  |   600 |  64 | 10 | 1050.00 |     15 |
|    2 | 1321  |   500 |  64 |  8 |  970.00 |     12 |
|    6 | 1298  |   450 |  64 | 10 |  950.00 |     12 |
|    1 | 1298  |   350 |  32 |  4 |  700.00 |     11 |
+------+-------+-------+-----+----+---------+--------+
6 rows in set (0.00 sec)
```

Break Time!

Next is `JOIN`... :/

Data from more than one Table

SKILLPILLS

- Last session, Jeremie described the “Normal Forms”
- We learned that often we should use multiple small tables
- But we still want to see different bits of data together!

code	model	speed	ram	hd	cd	price	maker	model	type
1	1232	500	64	5	12x	600.00	A	1232	PC
2	1121	750	128	14	40x	850.00	B	1121	PC
3	1233	500	64	5	12x	600.00	A	1233	PC
4	1121	600	128	14	40x	850.00	B	1121	PC
5	1121	600	128	8	40x	850.00	B	1121	PC
6	1233	750	128	20	50x	950.00	A	1233	PC
7	1232	500	32	10	12x	400.00	A	1232	PC
8	1232	450	64	8	24x	350.00	A	1232	PC
9	1232	450	32	10	24x	350.00	A	1232	PC
10	1260	500	32	10	12x	350.00	E	1260	PC
11	1233	900	128	40	40x	980.00	A	1233	PC
12	1233	800	128	20	50x	970.00	A	1233	PC

Multiple Table Selection (Cross Joins)

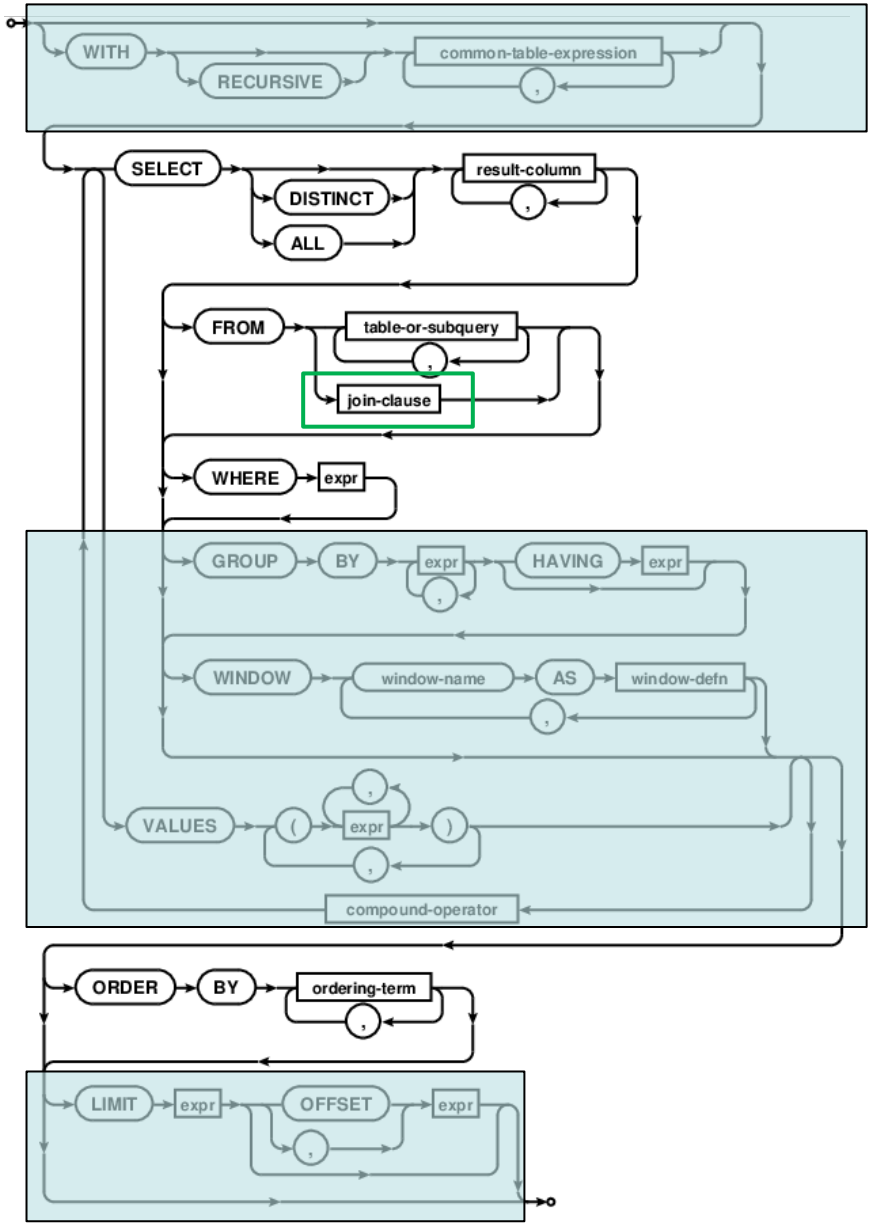
SKILLPILLS

Our first attempt to get data from multiple tables might be like one of these:

```
SELECT * FROM PC, Product;
```

```
SELECT * FROM PC, Product WHERE
PC.model=Product.model;
```

The JOIN Statement



- If we want to only get specific Records with matching elements, we use `JOIN`.
- We specify some criteria to `JOIN` the tables `ON`.

```
SELECT * FROM PC
JOIN Product ON
PC.model=Product.model;
```

JOIN Shortcuts

SKILLPILLS

- If we want to JOIN on several conditions, typing the table name repeatedly can be tedious
- We can use the "AS" again to 'rename' it for the operation
- We can use multiple lines to make it easier to read

```
SELECT PC.*, P.maker, P.type
FROM PC JOIN Product AS P
ON PC.model = P.model;
```

```
MariaDB [skill_pill]> select PC.*, P.maker, P.type
    -> FROM PC JOIN Product AS P
    -> ON PC.model = P.model;
+------+-------+-------+-----+----+-----+--------+-------+------+
| code | model | speed | ram | hd | cd  | price  | maker | type |
+------+-------+-------+-----+----+-----+--------+-------+------+
|    1 | 1232  |   500 |  64 |  5 | 12x | 600.00 | A     | PC   |
|    2 | 1121  |   750 | 128 | 14 | 40x | 850.00 | B     | PC   |
|    3 | 1233  |   500 |  64 |  5 | 12x | 600.00 | A     | PC   |
|    4 | 1121  |   600 | 128 | 14 | 40x | 850.00 | B     | PC   |
|    5 | 1121  |   600 | 128 |  8 | 40x | 850.00 | B     | PC   |
|    6 | 1233  |   750 | 128 | 20 | 50x | 950.00 | A     | PC   |
|    7 | 1232  |   500 |  32 | 10 | 12x | 400.00 | A     | PC   |
|    8 | 1232  |   450 |  64 |  8 | 24x | 350.00 | A     | PC   |
|    9 | 1232  |   450 |  32 | 10 | 24x | 350.00 | A     | PC   |
|   10 | 1260  |   500 |  32 | 10 | 12x | 350.00 | E     | PC   |
|   11 | 1233  |   900 | 128 | 40 | 40x | 980.00 | A     | PC   |
|   12 | 1233  |   800 | 128 | 20 | 50x | 970.00 | A     | PC   |
+------+-------+-------+-----+----+-----+--------+-------+------+
12 rows in set (0.00 sec)
```

```
MariaDB [skill_pill]> select PC.*, P.maker, P.type
    -> FROM PC JOIN Product AS P
    -> ON PC.model = P.model;
+------+-------+-------+-----+----+-----+--------+-------+------+
| code | model | speed | ram | hd | cd  | price  | maker | type |
+------+-------+-------+-----+----+-----+--------+-------+------+
|    1 | 1232  |   500 |  64 |  5 | 12x | 600.00 | A     | PC   |
|    2 | 1121  |   750 | 128 | 14 | 40x | 850.00 | B     | PC   |
|    3 | 1233  |   500 |  64 |  5 | 12x | 600.00 | A     | PC   |
|    4 | 1121  |   600 | 128 | 14 | 40x | 850.00 | B     | PC   |
|    5 | 1121  |   600 | 128 |  8 | 40x | 850.00 | B     | PC   |
|    6 | 1233  |   750 | 128 | 20 | 50x | 950.00 | A     | PC   |
|    7 | 1232  |   500 |  32 | 10 | 12x | 400.00 | A     | PC   |
|    8 | 1232  |   450 |  64 |  8 | 24x | 350.00 | A     | PC   |
|    9 | 1232  |   450 |  32 | 10 | 24x | 350.00 | A     | PC   |
|   10 | 1260  |   500 |  32 | 10 | 12x | 350.00 | E     | PC   |
|   11 | 1233  |   900 | 128 | 40 | 40x | 980.00 | A     | PC   |
|   12 | 1233  |   800 | 128 | 20 | 50x | 970.00 | A     | PC   |
+------+-------+-------+-----+----+-----+--------+-------+------+
12 rows in set (0.00 sec)
```

Exercises (Inner Join)

SKILLPILLS

1. Print all Laptops along with their Makers

```
+------+-------+-------+-----+-----+---------+--------+-------+
| code | model | speed | ram | hd | price    | screen | maker |
+------+-------+-------+-----+-----+---------+--------+-------+
```

2. Find the maker and speed of Laptops with a hard drive capacity of at least 10GB. Display the fields like this (you'll need quote marks for the spaces):

```
+-----+-----------+---------------------+
| Mkr | CPU Speed | Hard drive capacity |
+-----+-----------+---------------------+
```

3. Find the maker and model of each PC with a speed above 450 MHz

```
+-------+-------+------+-------+
| maker | model | type | speed |
+-------+-------+------+-------+
```

Types of JOIN

SKILLPILLS

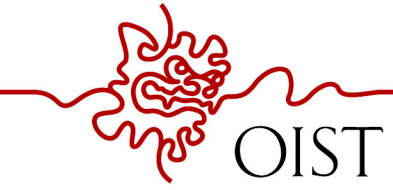
- There are several types of JOIN in SQL

- The types (as specified by ANSI SQL) are
  - Inner Join (this is the default, and most common type)
  - Left Outer Join (Left Join)
  - Right Outer Join (Right Join)
  - Full Outer Join (MySQL does not support this directly)
  - Cross Join

Left and Right (Outer) Joins

SKILLPILLS

- Left and Right Joins return all of the records from one table, and the matching records from the other table
- When there are no matching elements, NULL is returned
- Try this out with the PC and Product tables!
- The order of the tables is important (Left Join prioritizes the ‘left’ table, not the ‘joined’ table, Right Join is the opposite)



```
MariaDB [skill_pill]> select * from Product left join PC on Product.model=PC.model;
+-------+-------+---------+------+-------+-------+------+------+------+--------+
| maker | model | type    | code | model | speed | ram  | hd   | cd   | price  |
+-------+-------+---------+------+-------+-------+------+------+------+--------+
| B     | 1121  | PC      |    2 | 1121  |   750 |  128 |   14 | 40x  | 850.00 |
| B     | 1121  | PC      |    4 | 1121  |   600 |  128 |   14 | 40x  | 850.00 |
| B     | 1121  | PC      |    5 | 1121  |   600 |  128 |    8 | 40x  | 850.00 |
| A     | 1232  | PC      |    1 | 1232  |   500 |   64 |    5 | 12x  | 600.00 |
| A     | 1232  | PC      |    7 | 1232  |   500 |   32 |   10 | 12x  | 400.00 |
| A     | 1232  | PC      |    8 | 1232  |   450 |   64 |    8 | 24x  | 350.00 |
| A     | 1232  | PC      |    9 | 1232  |   450 |   32 |   10 | 24x  | 350.00 |
| A     | 1233  | PC      |    3 | 1233  |   500 |   64 |    5 | 12x  | 600.00 |
| A     | 1233  | PC      |    6 | 1233  |   750 |  128 |   20 | 50x  | 950.00 |
| A     | 1233  | PC      |   11 | 1233  |   900 |  128 |   40 | 40x  | 980.00 |
| A     | 1233  | PC      |   12 | 1233  |   800 |  128 |   20 | 50x  | 970.00 |
| E     | 1260  | PC      |   10 | 1260  |   500 |   32 |   10 | 12x  | 350.00 |
| A     | 1276  | Printer | NULL | NULL  |  NULL | NULL | NULL | NULL |   NULL |
| D     | 1288  | Printer | NULL | NULL  |  NULL | NULL | NULL | NULL |   NULL |
| A     | 1298  | Laptop  | NULL | NULL  |  NULL | NULL | NULL | NULL |   NULL |
| C     | 1321  | Laptop  | NULL | NULL  |  NULL | NULL | NULL | NULL |   NULL |
| A     | 1401  | Printer | NULL | NULL  |  NULL | NULL | NULL | NULL |   NULL |
| A     | 1408  | Printer | NULL | NULL  |  NULL | NULL | NULL | NULL |   NULL |
| D     | 1433  | Printer | NULL | NULL  |  NULL | NULL | NULL | NULL |   NULL |
| E     | 1434  | Printer | NULL | NULL  |  NULL | NULL | NULL | NULL |   NULL |
| B     | 1750  | Laptop  | NULL | NULL  |  NULL | NULL | NULL | NULL |   NULL |
| A     | 1752  | Laptop  | NULL | NULL  |  NULL | NULL | NULL | NULL |   NULL |
| E     | 2112  | PC      | NULL | NULL  |  NULL | NULL | NULL | NULL |   NULL |
| E     | 2113  | PC      | NULL | NULL  |  NULL | NULL | NULL | NULL |   NULL |
+-------+-------+---------+------+-------+-------+------+------+------+--------+
24 rows in set (0.00 sec)
```

UNION

SKILLPILLS

- You can get results from more than one SELECT in the same result set
- Use UNION to do this

```
SELECT speed from PC
UNION
SELECT speed from Laptop;
```

Union + Join

SKILLPILLS

- Use a combination of UNION and JOIN to find model number, type and speed for all Laptops and PCs

- Take care that several parameters are not unique! To ensure that even 'duplicate' rows are returned, use `UNION ALL` instead of `UNION`.

Carrying out operations on results

SKILLPILLS

- Several functions are defined by typical DBMSs
- One of these (for MySQL) is `AVG`
- Use `UNION` together with `AVG` and `JOIN` to get the average speeds of Laptops and PCs

```
+-------------+--------+
| avg(speed) | type    |
+-------------+--------+
|    608.3333 | PC      |
|    566.6667 | Laptop  |
+-------------+--------+
2 rows in set (0.01 sec)
```