SQL

Basic Syntax

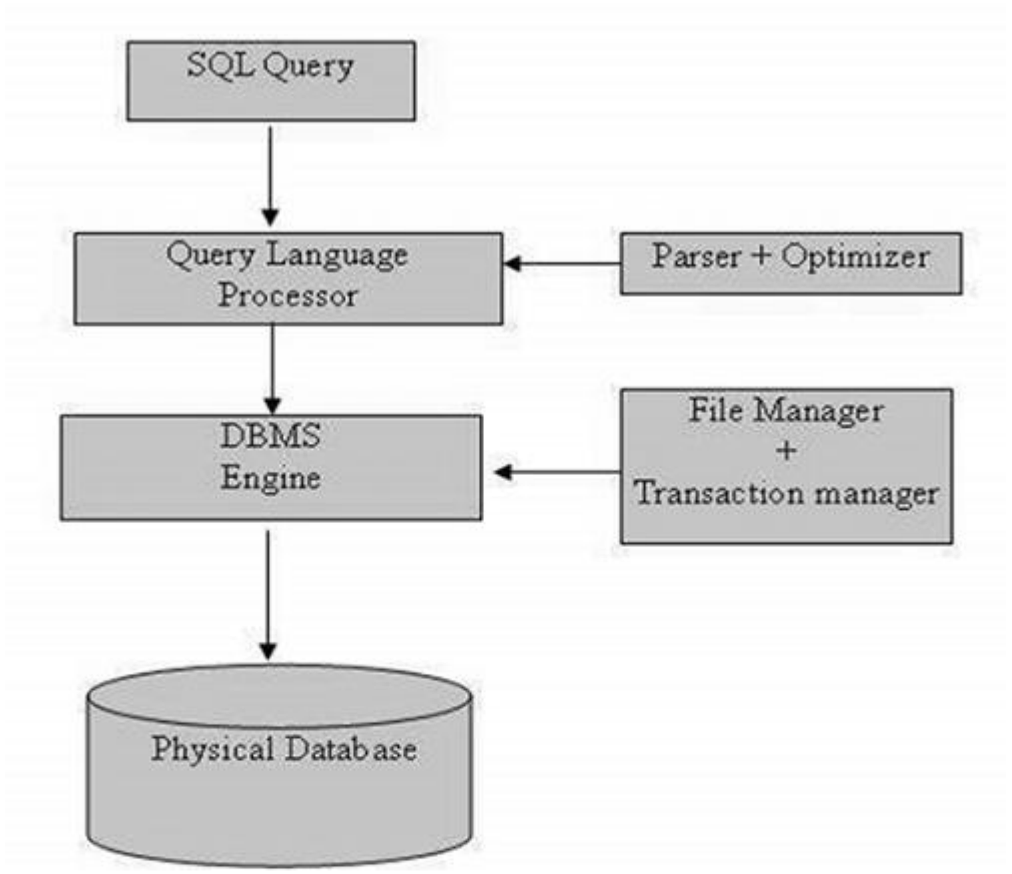
SQL

- SQL is a language to operate databases
- SQL is an **ANSI** (American National Standards Institute) standard language
- SQL is Structured Query Language,
  - a computer language for storing, manipulating and retrieving data stored in a relational database.
- SQL is the standard language for Relational Database System
  - like MySQL, MS Access, Oracle, Sybase, Informix, Postgres and SQL Server

WHY SQL?

- Allows users to **access** data in the relational database management systems.
- Allows users to **describe** the data.
- Allows users to **define** the data in a database and **manipulate** that data.
- Allows to **embed within other languages** using SQL modules, libraries & pre-compilers.
- Allows users to **create** and **drop** databases and tables.
- Allows users to create **view, stored procedure, functions** in a database.
- Allows users to set **permissions** on tables, procedures and views.

SQL Process


SQL Query
Query Language Processor
Parser + Optimizer
DBMS Engine
File Manager + Transaction manager
Physical Database

SQL Commands

- DDL - Data Definition Language

No.	Command & Description
1	**CREATE** Creates a new table, a view of a table, or other object in the database.
2	**ALTER** Modifies an existing database object, such as a table.
3	**DROP** Deletes an entire table, a view of a table or other objects in the database.

SQL Commands

- DML - Data Manipulation Language

No.	Command & Description
1	**SELECT** Retrieves certain records from one or more tables.
2	**INSERT** Creates a record.
3	**UPDATE** Modifies records.
4	**DELETE** Deletes records.

SQL Commands

- DCL - Data Control Language

No.	Command & Description
1	**GRANT** Gives a privilege to user.
2	**REVOKE** Takes back privileges granted from user.

SQL Data Types

- Character string data type
  - CHAR(n)
  - VARCHAR(n)
- Date and Time types
  - DATE
  - TIME
  - TIMESTAMP
- SQL numeric data types:
  - BIT(n)
  - BIT VARYING (n)
  - DECIMAL (p,s)
  - INTEGER
  - SMALLINT
  - BIGINT
  - FLOAT(p,s)
  - DOUBLE PRECISION (p,s)
  - REAL(s)

Exact Numeric Data Types

DATA TYPE	FROM	TO
bigint	-9,223,372,036,854,775,808	9,223,372,036,854,775,807
int	-2,147,483,648	2,147,483,647
smallint	-32,768	32,767
tinyint	0	255
bit	0	1
decimal	-10^38 +1	10^38 -1
numeric	-10^38 +1	10^38 -1

Approximate Numeric Data Types

DATA TYPE	FROM	TO
float	-1.79E + 308	1.79E + 308
real	-3.40E + 38	3.40E + 38

Date and Time Data Types

DATA TYPE	FROM	TO
datetime	Jan 1, 1753	Dec 31, 9999
smalldatetime	Jan 1, 1900	Jun 6, 2079
date	Stores a date like June 30, 1991	
time	Stores a time of day like 12:30 P.M.	

Character Strings Data Types

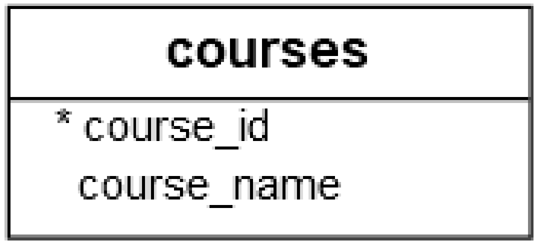
DATA TYPE & Description
char Maximum length of 8,000 characters.(Fixed length non-Unicode characters)
varchar Maximum of 8,000 characters.(Variable-length non-Unicode data).
varchar(max) Maximum length of 2E + 31 characters, Variable-length non-Unicode data (SQL Server 2005 only).
text Variable-length non-Unicode data with a maximum length of 2,147,483,647 characters.

SQL Auto Increment

```
CREATE TABLE leave_requests (
   request_id INT AUTO_INCREMENT,
   employee_id INT NOT NULL,
   start_date DATE NOT NULL,
   end_date DATE NOT NULL,
   leave_type INT NOT NULL,
   PRIMARY KEY(request_id)
);
```

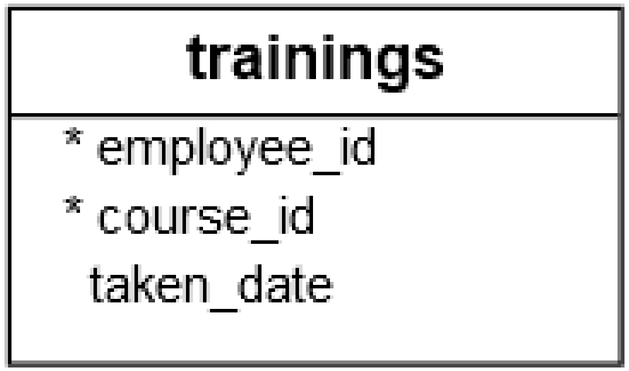
SQL CREATE TABLE

```
CREATE TABLE table_name (
    column_name_1 data_type default value column_constraint,
    column_name_2 data_type default value column_constraint,
    column_name_3 data_type default value column_constraint,
    ...,
    table_constraint
);
```



```
CREATE TABLE courses (
   course_id INT AUTO_INCREMENT PRIMARY KEY,
   course_name VARCHAR(50) NOT NULL
);
```

SQL CREATE TABLE



```sql
CREATE TABLE trainings (
    employee_id INT,
    course_id INT,
    taken_date DATE,
    PRIMARY KEY (employee_id , course_id)
);
```

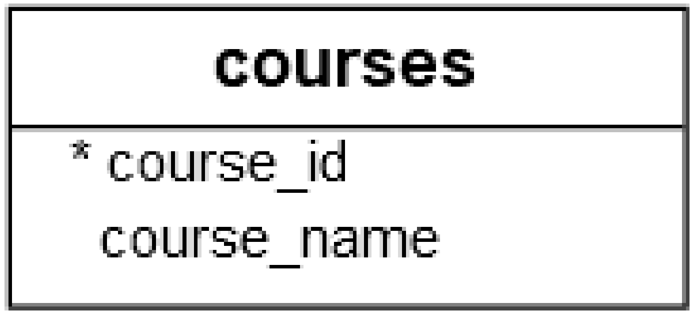
SQL ALTER TABLE ADD column

ALTER TABLE table_name

ADD new_colum data_type column_constraint [AFTER existing_column];

SQL ALTER TABLE ADD column

```
ALTER TABLE courses ADD credit_hours INT NOT
NULL;
```



```
ALTER TABLE courses
ADD fee NUMERIC (10, 2) AFTER course_name,
ADD max_limit INT AFTER course_name;
```

SQL ALTER TABLE MODIFY column

```
ALTER TABLE table_name
MODIFY column_definition;
```

Try this:

```
ALTER TABLE courses
MODIFY fee NUMERIC (10, 2) NOT NULL;
```

SQL ALTER TABLE DROP column

```
ALTER TABLE table_name
DROP column_name,
DROP colum_name,
```

Try this:

```
ALTER TABLE courses DROP COLUMN fee;
ALTER TABLE courses
DROP COLUMN max_limit,
DROP COLUMN credit_hours;
```

SQL ALTER TABLE MODIFY column

```
ALTER TABLE table_name
MODIFY column_definition;
```

SQL CONSTRAINTS

- SQL Primary Key
- SQL Foreign Key
- SQL NOT NULL Constraint

SQL Primary Key

```sql
CREATE TABLE projects (
    project_id INT PRIMARY KEY,
    project_name VARCHAR(255),
    start_date DATE NOT NULL,
    end_date DATE NOT NULL
);
```

```sql
CREATE TABLE projects (
    project_id INT,
    project_name VARCHAR(255),
    start_date DATE NOT NULL,
    end_date DATE NOT NULL,
    CONSTRAINT pk_id PRIMARY KEY (project_id)
);
```

```sql
ALTER TABLE project_milestones
ADD CONSTRAINT pk_milestone_id PRIMARY KEY (milestone_id);
```

SQL Foreign Key

```sql
CREATE TABLE project_milestones (
    milestone_id INT AUTO_INCREMENT PRIMARY KEY,
    project_id INT,
    milestone_name VARCHAR(100),
    FOREIGN KEY (project_id)
        REFERENCES projects (project_id)
);
```

```sql
ALTER TABLE project_milestones
ADD CONSTRAINT fk_project FOREIGN KEY(project_id)
  REFERENCES projects(project_id);
```

Exercise

My_Company Database

USER & GRANT

- Untuk menjalankan sebuah database, user perlu diberi hak akses (GRANT)
- Syntax:

```
CREATE USER "username" IDENTIFIED BY "password” ;

GRANT ALL PRIVILEGES ON databasename.* TO ‘username'@'%';

GRANT
    SELECT, INSERT, UPDATE, DELETE, CREATE, DROP,
    INDEX, ALTER, CREATE TEMPORARY TABLES, CREATE
    VIEW, EVENT, TRIGGER, SHOW VIEW, EXECUTE
ON databasename.*
TO ‘username'@’%’ ;
```

ROLE

- ROLE diperlukan bila beberapa user memiliki hak akses yang berbeda-beda.
- Misal pada database toko: kasir, manager, owner
- Syntax CREATE dan GRANT:

```
CREATE ROLE "kasir“, “manager” ;

GRANT insert, update, select ON database.table_name TO kasir

GRANT kasir TO ansisa, anna, dina
```

Exercises

Buat user baru dengan hak akses ke database MyCompany

SQL – CREATE & DROP Database

Syntax:

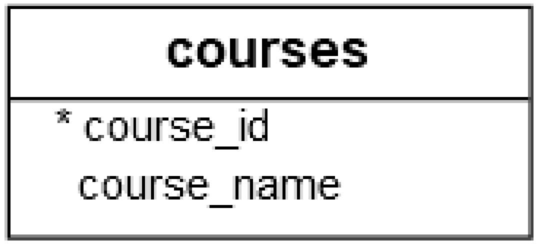
```
CREATE DATABASE DatabaseName;

DROP DATABASE DatabaseName
```

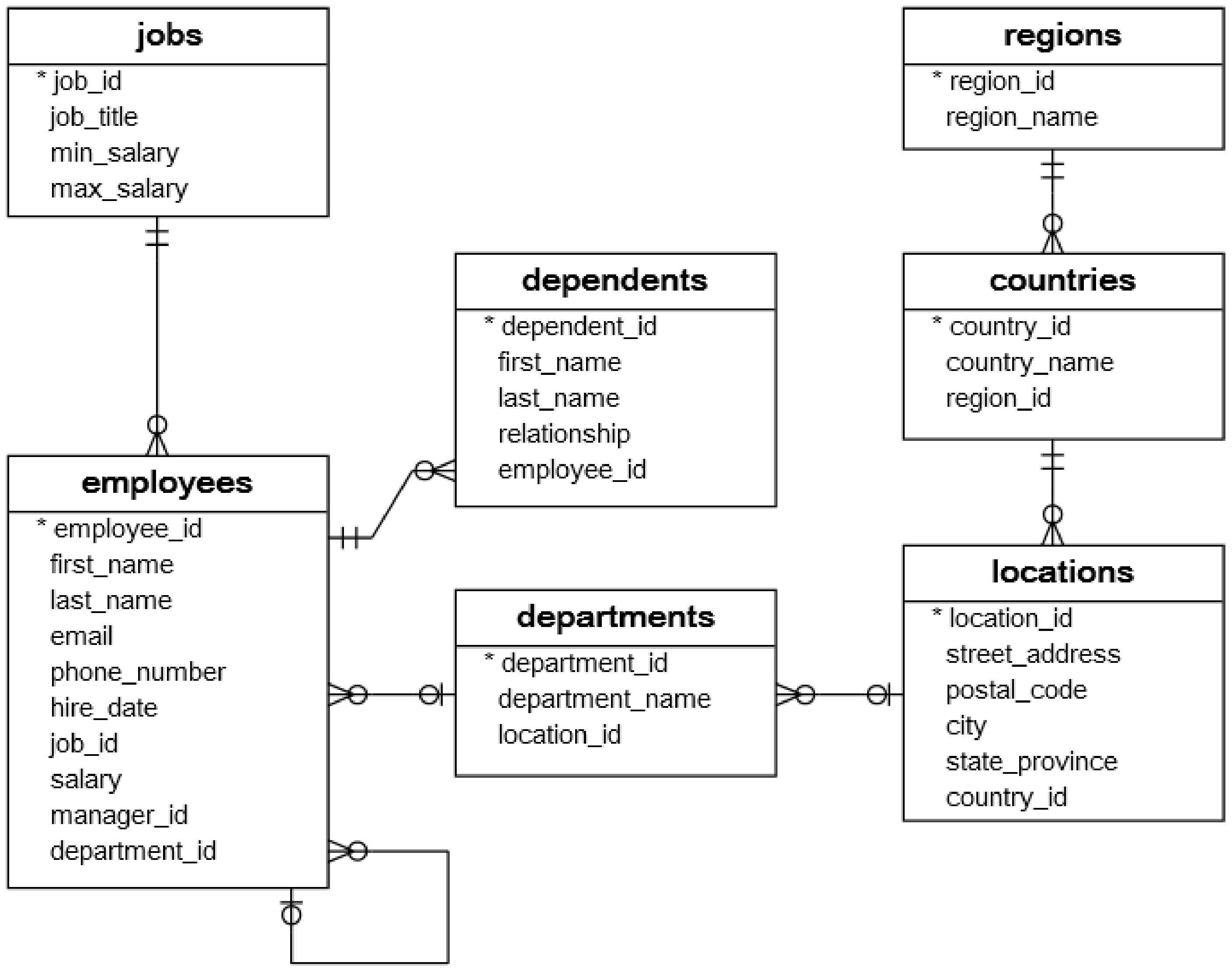
SQL CREATE TABLE

```
CREATE TABLE table_name (
    column_name_1 data_type default value column_constraint,
    column_name_2 data_type default value column_constraint,
    column_name_3 data_type default value column_constraint,
    ...,
    PRIMARY KEY( one or more columns )
    other table_constraint
);
```

```
CREATE TABLE courses (
    course_id INT AUTO_INCREMENT PRIMARY KEY,
    course_name VARCHAR(50) NOT NULL
);
```



Create all tables for this database



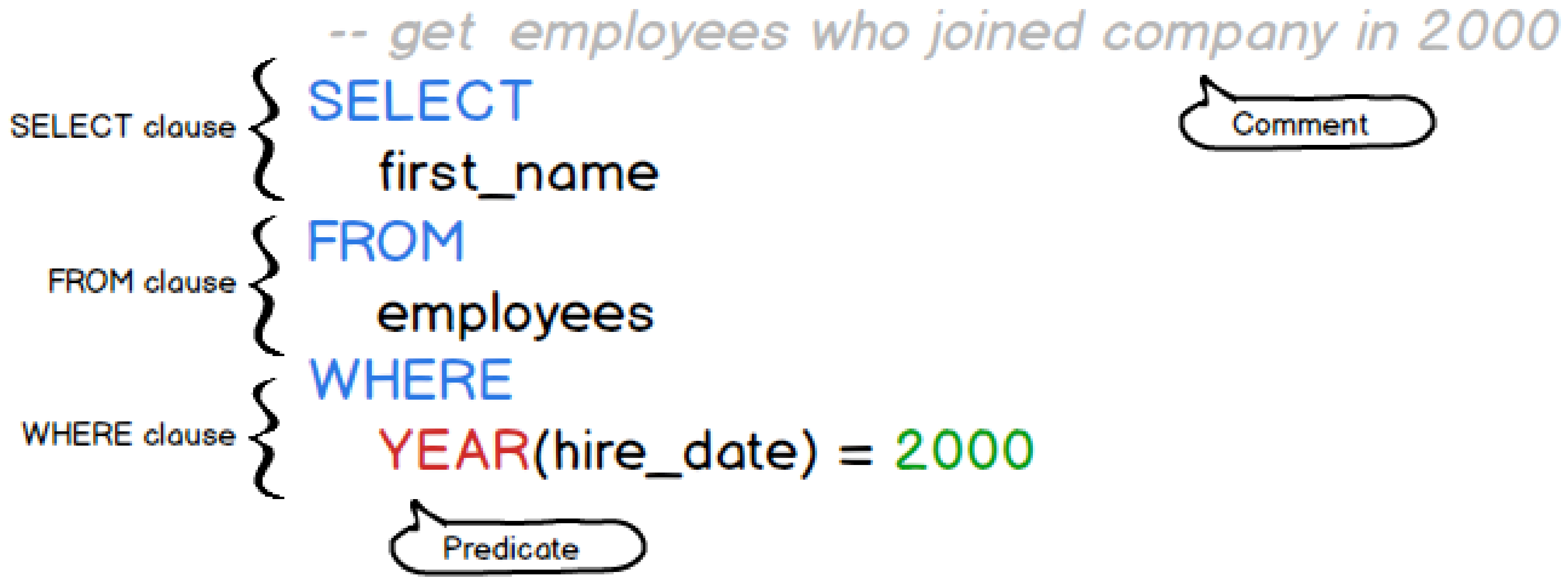
SQL INSERT

Syntax:

```
INSERT INTO tablename (column1, column2,...columnN)
VALUES (value1, value2,...valueN);
```

Insert all data MyCompany into the tables

SQL SELECT



```
-- get  employees who joined company in 2000
SELECT
    first_name
FROM
    employees
WHERE
    YEAR(hire_date) = 2000
```

SQL WHERE Clause

Condition

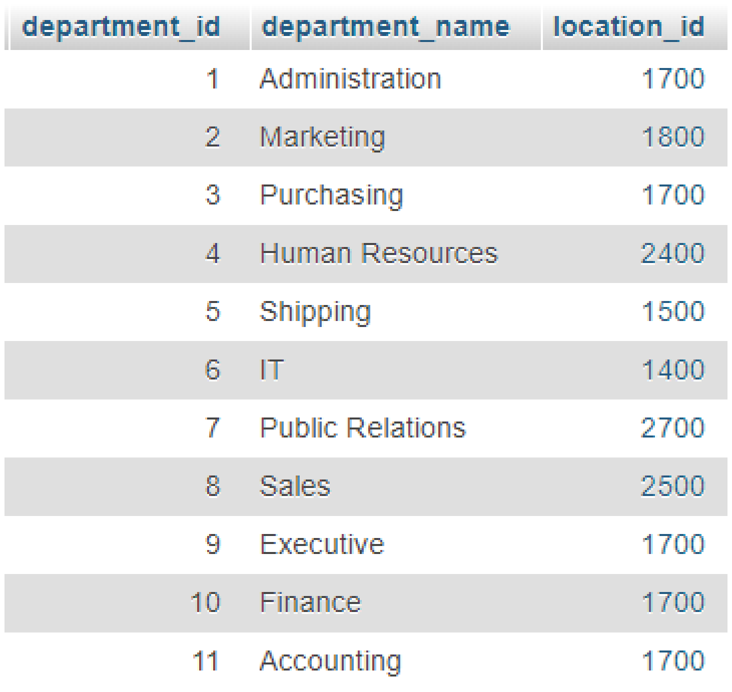
```sql
SELECT first_name, last_name
FROM employees
WHERE department_id = 1400;
```

Condition

```sql
DELETE FROM employees
WHERE hire_date < '1990-01-01‘ ;
```

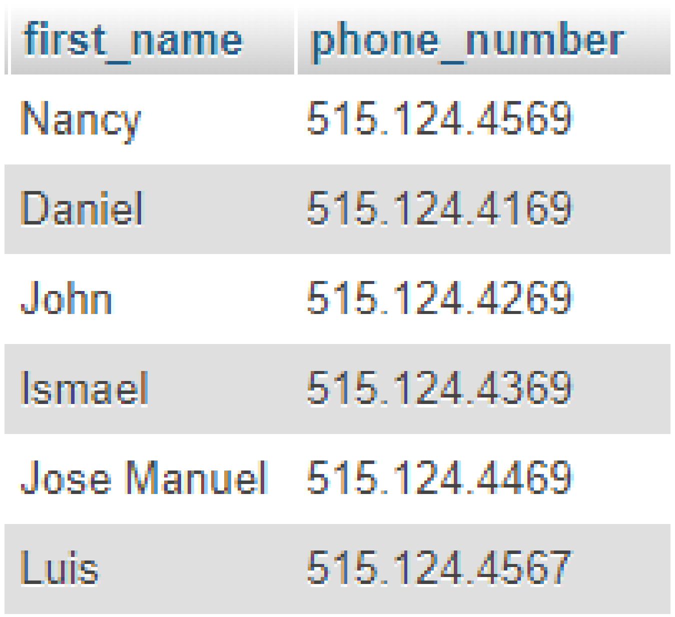
Exercises

1. Tampilkan data semua departemen yang ada



department_id	department_name	location_id
1	Administration	1700
2	Marketing	1800
3	Purchasing	1700
4	Human Resources	2400
5	Shipping	1500
6	IT	1400
7	Public Relations	2700
8	Sales	2500
9	Executive	1700
10	Finance	1700
11	Accounting	1700

2. Tampilkan nama dan telefon karyawan yang ada di departemen dengan no ID = 10



first_name	phone_number
Nancy	515.124.4569
Daniel	515.124.4169
John	515.124.4269
Ismael	515.124.4369
Jose Manuel	515.124.4469
Luis	515.124.4567

SQL AND and OR Operators

SELECT column1, column2, columnN

FROM table_name

WHERE [condition1]

AND|OR [condition2]

...

AND|OR [conditionN];

pilih AND atau OR

SQL Comparison Operators

Operator	Meaning
=	Equal
<>	Not equal to
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to

SQL Server STRING Function

Function	**Description**
CHARINDEX	Returns the position of a substring in a string
CONCAT	Adds two or more strings together
LEN	Returns the length of a string
LOWER	Converts a string to lower-case
LTRIM	Removes leading spaces from a string
REPLACE	Replaces all occurrences of a substring within a string, with a new substring
REVERSE	Reverses a string and returns the result
RTRIM	Removes trailing spaces from a string
STR	Returns a number as string
SUBSTRING	Extracts some characters from a string
TRIM	Removes leading and trailing spaces (or other specified characters) from a string
UPPER	Converts a string to upper-case

SQL Logical Operators

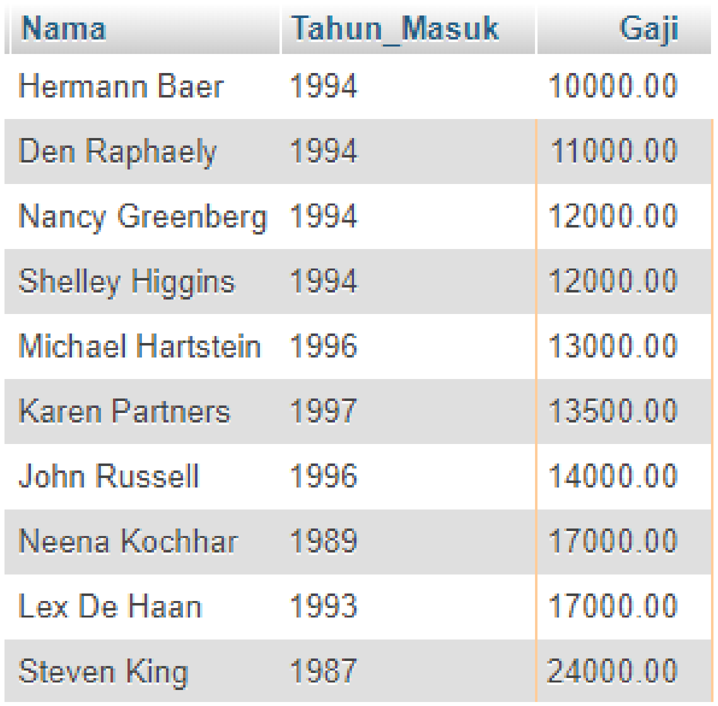
Operator	Meaning
ALL	Return true if all comparisons are true
AND	Return true if both expressions are true
ANY	Return true if any one of the comparisons is true.
BETWEEN	Return true if the operand is within a range
EXISTS	Return true if a subquery contains any rows
IN	Return true if the operand is equal to one of the value in a list
LIKE	Return true if the operand matches a pattern
NOT	Reverse the result of any other Boolean operator.
OR	Return true if either expression is true
SOME	Return true if some of the expressions are true

SQL Server STRING Function (complete)

Function	Description
ASCII	Returns the ASCII value for the specific character
CHAR	Returns the character based on the ASCII code
CHARINDEX	Returns the position of a substring in a string
CONCAT	Adds two or more strings together
Concat with +	Adds two or more strings together
CONCAT_WS	Adds two or more strings together with a separator
DATALENGTH	Returns the number of bytes used to represent an expression
DIFFERENCE	Compares two SOUNDEX values, and returns an integer value
FORMAT	Formats a value with the specified format
LEFT	Extracts a number of characters from a string (starting from left)
LEN	Returns the length of a string
LOWER	Converts a string to lower-case
LTRIM	Removes leading spaces from a string
NCHAR	Returns the Unicode character based on the number code
PATINDEX	Returns the position of a pattern in a string
QUOTENAME	Returns a Unicode string with delimiters added to make the string a valid SQL Server delimited identifier
REPLACE	Replaces all occurrences of a substring within a string, with a new substring
REPLICATE	Repeats a string a specified number of times
REVERSE	Reverses a string and returns the result
RIGHT	Extracts a number of characters from a string (starting from right)
RTRIM	Removes trailing spaces from a string
SOUNDEX	Returns a four-character code to evaluate the similarity of two strings
SPACE	Returns a string of the specified number of space characters
STR	Returns a number as string
STUFF	Deletes a part of a string and then inserts another part into the string, starting at a specified position
SUBSTRING	Extracts some characters from a string
TRANSLATE	Returns the string from the first argument after the characters specified in the second argument are translated into the characters specified in the third argument.
TRIM	Removes leading and trailing spaces (or other specified characters) from a string
UNICODE	Returns the Unicode value for the first character of the input expression
UPPER	Converts a string to upper-case

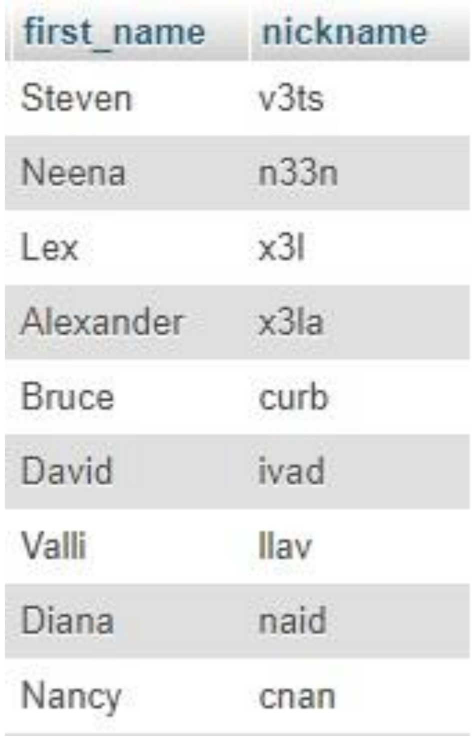
Exercises

3. Tampilkan semua data employees berikut: nama lengkap, tahun masuk kerja, dan gaji dengan gaji > 9.000



Nama	Tahun_Masuk	Gaji
Hermann Baer	1994	10000.00
Den Raphaely	1994	11000.00
Nancy Greenberg	1994	12000.00
Shelley Higgins	1994	12000.00
Michael Hartstein	1996	13000.00
Karen Partners	1997	13500.00
John Russell	1996	14000.00
Neena Kochhar	1989	17000.00
Lex De Haan	1993	17000.00
Steven King	1987	24000.00

4. Tampilkan 4 huruf pertama dari firstname, ubah jadi huruf kecil semua, ubah semua huruf "e" menjadi angka "3" lalu *reverse* ke-4 huruf tersebut



first_name	nickname
Steven	v3ts
Neena	n33n
Lex	x3l
Alexander	x3la
Bruce	curb
David	ivad
Valli	llav
Diana	naid
Nancy	cnan

ANSWER

```
SELECT
   concat (first_name," ",last_name) AS Nama,
   date_format(hire_date,"%Y") AS Tahun_Masuk,
   salary AS Gaji
FROM `employees`
WHERE salary > 9000
```

ANSWER

```
SELECT
   first_name,
   REPLACE(
       REVERSE(
               LOWER(
                  SUBSTRING(first_name,1,4)
               )
        )
   ,'e','3') as nickname
FROM `employees` ;
```

aggregate functions

- SUM([ALL | DISTINCT] expression)
- AVG([ALL | DISTINCT] expression)
- COUNT([ALL | DISTINCT] expression)
- COUNT(*)
- MAX(expression)
- MIN(expression)

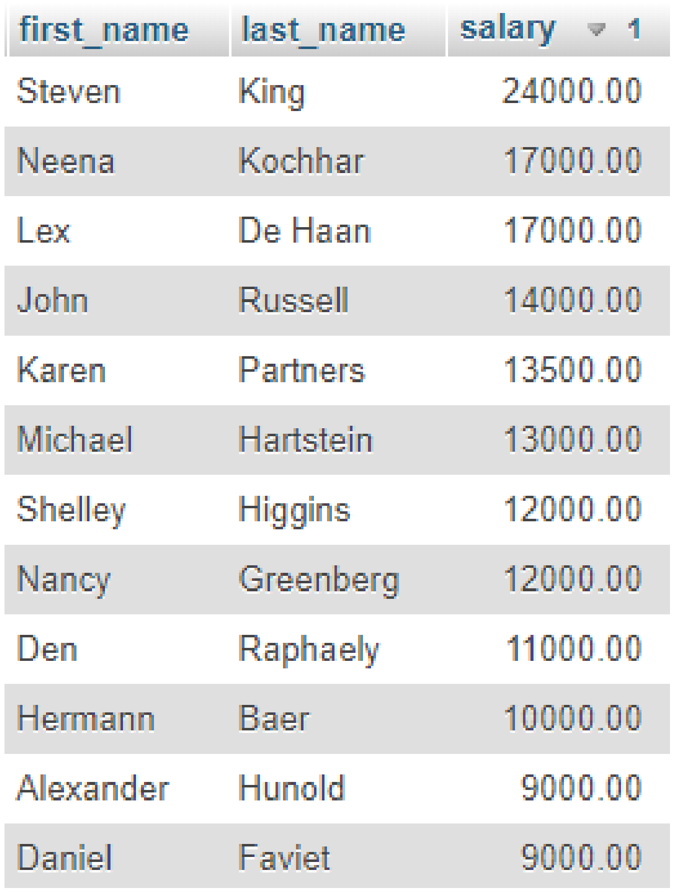
GROUP BY & ORDER BY

Syntax:

```
SELECT COUNT (*)
FROM tablename
GROUP BY column
ORDER BY column;
```

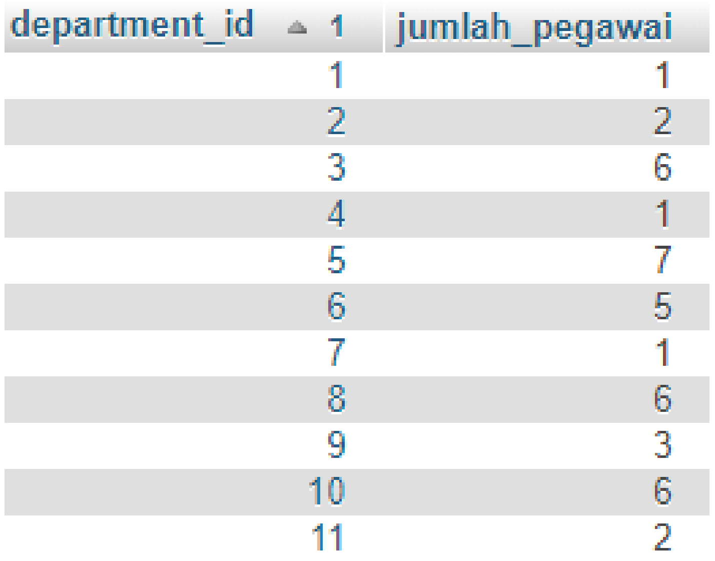
Exercises

5. Tampilkan data employees diurutkan dari gaji yang terbesar



first_name	last_name	salary ▼ 1
Steven	King	24000.00
Neena	Kochhar	17000.00
Lex	De Haan	17000.00
John	Russell	14000.00
Karen	Partners	13500.00
Michael	Hartstein	13000.00
Shelley	Higgins	12000.00
Nancy	Greenberg	12000.00
Den	Raphaely	11000.00
Hermann	Baer	10000.00
Alexander	Hunold	9000.00
Daniel	Faviet	9000.00

6. Tampilkan jumlah pegawai di setiap department, diurutkan dari nomor department terkecil



department_id ▲ 1	jumlah_pegawai
1	1
2	2
3	6
4	1
5	7
6	5
7	1
8	6
9	3
10	6
11	2

ANSWER

6. Tampilkan jumlah pegawai di setiap department, diurutkan dari nomor department terkecil

```
SELECT department_id, COUNT(*) as jumlah_pegawai
FROM `employees`
GROUP by department_id
ORDER BY department_id
```

SQL Syntax for >1 tables

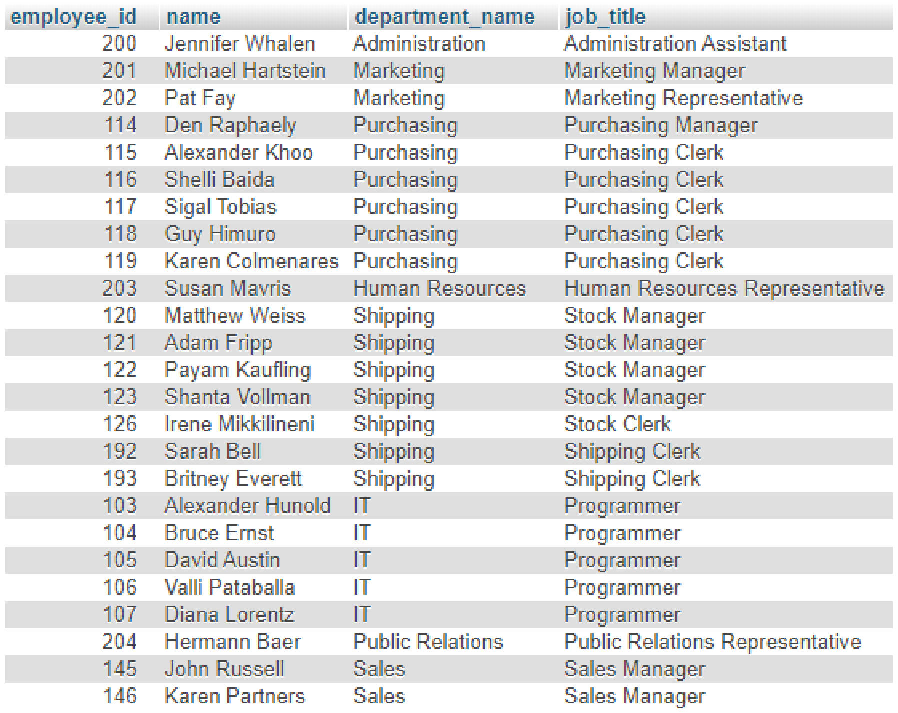
```
SELECT  t1.column11, t1.column12, t2.column21
FROM tableOne t1, tableTwo t2
WHERE  t1.columnRef = t2.columnRef
AND|OR [condition2]
...
AND|OR [conditionN];
```

Choose the table

Main condition

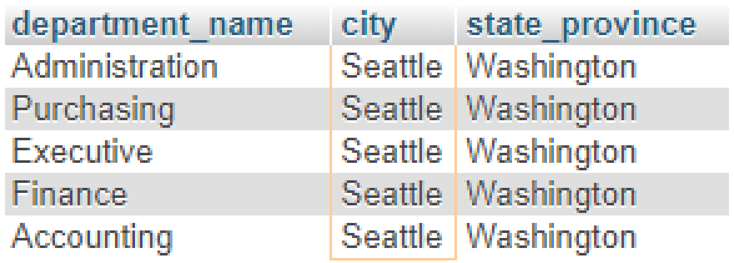
Exercises

7. Tampilkan data berikut: nama lengkap karyawan, nama departemen, job setiap karyawan



employee_id	name	department_name	job_title
200	Jennifer Whalen	Administration	Administration Assistant
201	Michael Hartstein	Marketing	Marketing Manager
202	Pat Fay	Marketing	Marketing Representative
114	Den Raphaely	Purchasing	Purchasing Manager
115	Alexander Khoo	Purchasing	Purchasing Clerk
116	Shelli Baida	Purchasing	Purchasing Clerk
117	Sigal Tobias	Purchasing	Purchasing Clerk
118	Guy Himuro	Purchasing	Purchasing Clerk
119	Karen Colmenares	Purchasing	Purchasing Clerk
203	Susan Mavris	Human Resources	Human Resources Representative
120	Matthew Weiss	Shipping	Stock Manager
121	Adam Fripp	Shipping	Stock Manager
122	Payam Kaufling	Shipping	Stock Manager
123	Shanta Vollman	Shipping	Stock Manager
126	Irene Mikkilineni	Shipping	Stock Clerk
192	Sarah Bell	Shipping	Shipping Clerk
193	Britney Everett	Shipping	Shipping Clerk
103	Alexander Hunold	IT	Programmer
104	Bruce Ernst	IT	Programmer
105	David Austin	IT	Programmer
106	Valli Pataballa	IT	Programmer
107	Diana Lorentz	IT	Programmer
204	Hermann Baer	Public Relations	Public Relations Representative
145	John Russell	Sales	Sales Manager
146	Karen Partners	Sales	Sales Manager

8. Tampilkan semua department yang ada di kota Seattle



department_name	city	state_province
Administration	Seattle	Washington
Purchasing	Seattle	Washington
Executive	Seattle	Washington
Finance	Seattle	Washington
Accounting	Seattle	Washington

ANSWER

7. Tampilkan data berikut: nama lengkap karyawan, nama departemen, job setiap karyawan

```
SELECT
    employee_id,
    concat(first_name, " ", last_name) AS name,
    department_name,
    job_title
FROM
    employees e , departments d , jobs j
WHERE
    e.department_id = d.department_id
AND
    e.job_id = j.job_id
;
```

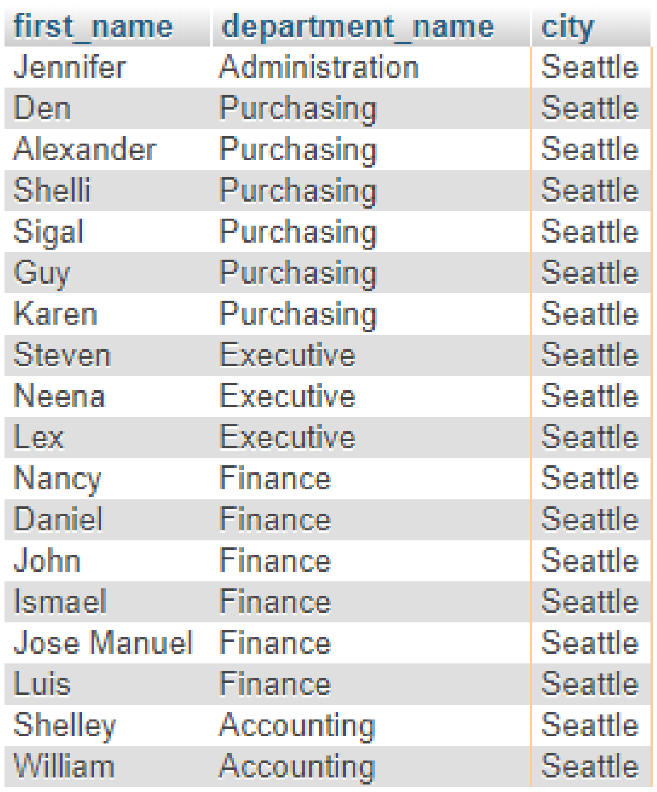
ANSWER

8. Tampilkan semua department yang ada di kota Seattle

```
SELECT department_name, city, state_province
FROM departments,  locations
WHERE departments.location_id = locations.location_id
AND city LIKE "seattle"
```

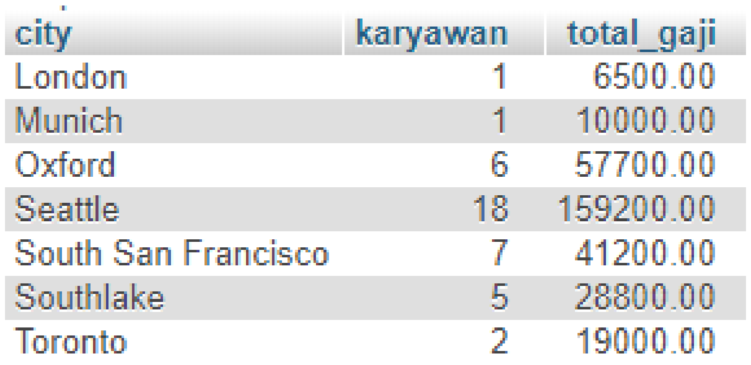
Exercises

9. Tampilkan nama, department dan kota untuk karyawan yang bekerja di kota Seattle



first_name	department_name	city
Jennifer	Administration	Seattle
Den	Purchasing	Seattle
Alexander	Purchasing	Seattle
Shelli	Purchasing	Seattle
Sigal	Purchasing	Seattle
Guy	Purchasing	Seattle
Karen	Purchasing	Seattle
Steven	Executive	Seattle
Neena	Executive	Seattle
Lex	Executive	Seattle
Nancy	Finance	Seattle
Daniel	Finance	Seattle
John	Finance	Seattle
Ismael	Finance	Seattle
Jose Manuel	Finance	Seattle
Luis	Finance	Seattle
Shelley	Accounting	Seattle
William	Accounting	Seattle

10. Tampilkan jumlah karyawan dan jumlah total gaji yang bekerja di masing-masing kota



city	karyawan	total_gaji
London	1	6500.00
Munich	1	10000.00
Oxford	6	57700.00
Seattle	18	159200.00
South San Francisco	7	41200.00
Southlake	5	28800.00
Toronto	2	19000.00

ANSWER

9. Tampilkan data berikut: nama lengkap karyawan, nama departemen, job setiap karyawan

```
SELECT first_name, department_name, city
FROM locations loc, departments dep, employees emp
WHERE loc.location_id = dep.location_id
AND dep.department_id = emp.department_id\
AND city LIKE "seattle"
```

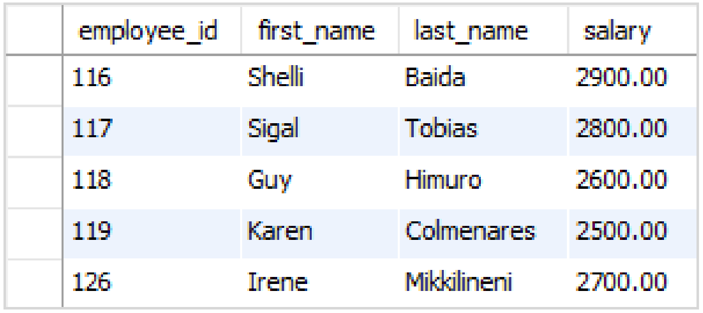
ANSWER

10. Tampilkan jumlah karyawan dan jumlah total gaji yang bekerja di masing-masing kota

```
SELECT
    city,
    COUNT(first_name) as karyawan,
    SUM(salary) as total_gaji
FROM
    locations l, departments d, employees e
WHERE
    l.location_id = d.location_id
    AND d.department_id = e.department_id
GROUP BY city
```

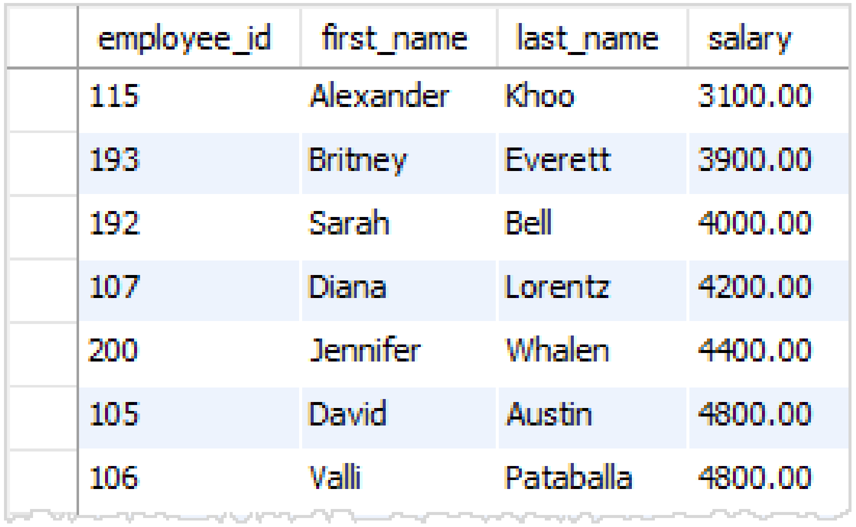
Exercises

11. find all employees whose salaries are BETWEEN 2,500 and 2,900



employee_id	first_name	last_name	salary
116	Shelli	Baida	2900.00
117	Sigal	Tobias	2800.00
118	Guy	Himuro	2600.00
119	Karen	Colmenares	2500.00
126	Irene	Mikkilineni	2700.00

12. find all employees whose salary are not in the range of 2,500 and 2,900. Use the NOT BETWEEN operator in the WHERE clause



employee_id	first_name	last_name	salary
115	Alexander	Khoo	3100.00
193	Britney	Everett	3900.00
192	Sarah	Bell	4000.00
107	Diana	Lorentz	4200.00
200	Jennifer	Whalen	4400.00
105	David	Austin	4800.00
106	Valli	Pataballa	4800.00

ANSWER

11. find all employees whose salaries are BETWEEN 2,500 and 2,900

```
SELECT
    employee_id, first_name, last_name, salary
FROM
    employees
WHERE
    salary BETWEEN 2500 AND 2900;
```

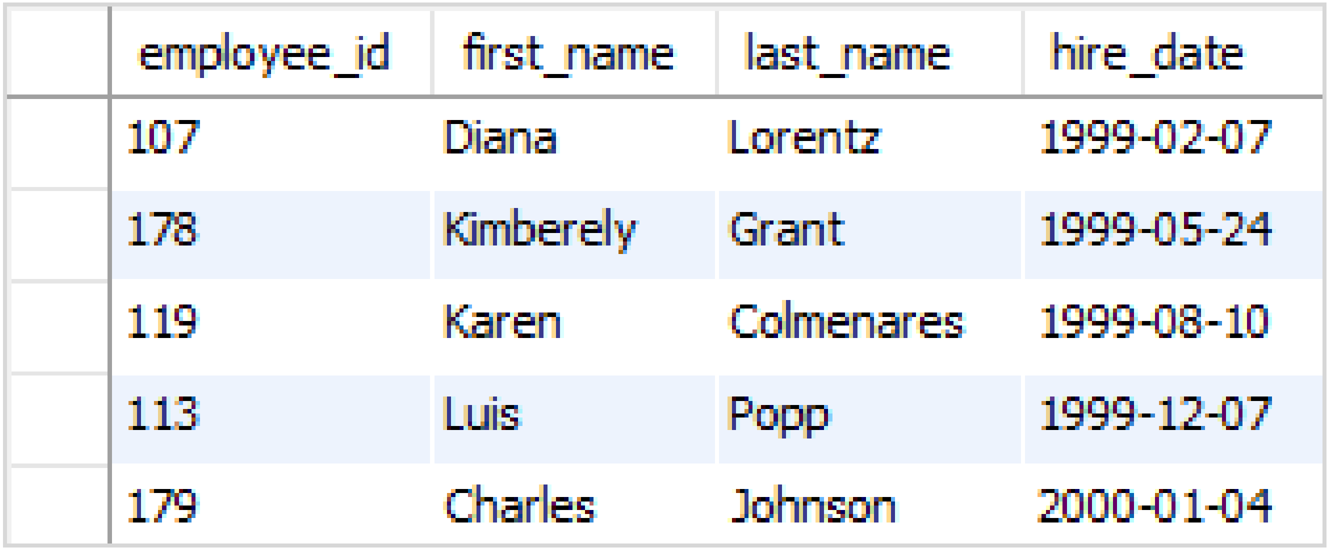
ANSWER

12. find all employees whose salary are not in the range of 2,500 and 2,900. Use the NOT BETWEEN operator in the WHERE clause

```
SELECT
    employee_id, first_name, last_name, salary
FROM
    employees
WHERE
    salary NOT BETWEEN 2500 AND 2900
ORDER BY salary;
```

Exercises

13. find all employees who joined the company between January 1, 1999, and December 31, 2000



employee_id	first_name	last_name	hire_date
107	Diana	Lorentz	1999-02-07
178	Kimberely	Grant	1999-05-24
119	Karen	Colmenares	1999-08-10
113	Luis	Popp	1999-12-07
179	Charles	Johnson	2000-01-04

ANSWER

13. find all employees who joined the company between January 1, 1999, and December 31, 2000

```
SELECT
    employee_id, first_name, last_name, hire_date
FROM
    employees
WHERE
    hire_date BETWEEN '1999-01-01' AND '2000-12-31'
ORDER BY hire_date;
```