Table Structures

CREATE TABLE employees (

employee\_id INT PRIMARY KEY,

name VARCHAR(100),

department\_id INT,

salary DECIMAL(10, 2),

hire\_date DATE

);

CREATE TABLE sales (

sale\_id INT PRIMARY KEY,

employee\_id INT,

sale\_date DATE,

amount DECIMAL(10, 2)

);

CREATE TABLE departments (

department\_id INT PRIMARY KEY,

department\_name VARCHAR(100)

);

CREATE TABLE products (

product\_id INT PRIMARY KEY,

product\_name VARCHAR(100),

category VARCHAR(100),

price DECIMAL(10, 2)

);

CREATE TABLE orders (

order\_id INT PRIMARY KEY,

customer\_id INT,

product\_id INT,

order\_date DATE,

quantity INT

);

CREATE TABLE customers (

customer\_id INT PRIMARY KEY,

name VARCHAR(100),

city VARCHAR(100)

);

Question (Write Query)

Write single query for all below questions

1. Find the second highest salary in each department.

Ans: **Select employees.department\_id, max(employees.employee\_salary) as Max\_emp\_sal from employees where employees.employee\_salary < (SELECT max(tb2.employee\_salary) from employees as tb2 where tb2.department\_id = employees.department\_id) group by employees.department\_id;**

1. Calculate the difference in sales between each employee's current sale and their previous sale.
2. Retrieve the average salary of employees in each department and the difference of each employee's salary from the department's average salary.
3. Find the cumulative percentage of total sales amount for each employee.

**Ans: SELECT employees.employee\_name, sales.employee\_id, `sale\_date`, sum(`amount`) as price FROM `sales` LEFT join employees on employees.id = sales.employee\_id group by sales.employee\_id,sales.sale\_date;**

1. Retrieve the top 3 most expensive products in each category.  
   **Ans: SELECT `category`, max(`price`) as expensive FROM `products` GROUP by `category`;**
2. Calculate the year-to-date (YTD) sales for each employee.
3. Retrieve the sales of each employee as a percentage of the total sales in their department.
4. Retrieve the name and total sales amount for the employee with the second highest total sales.
5. Find the average sales amount per employee for each department, including departments with no sales.
6. Retrieve the names of employees who have the highest sale amount in each department.
7. Identify the customers who have not placed any orders in the last 6 months.
8. List customers who have placed more orders than the average number of orders per customer.
9. Find the names of employees who have not sold any products in the "Electronics" category.
10. Write a single query using below mentioned tables to fetch list of samples with parent test available under the sample and child tests available under that parent test. If not child test available under a parent then child test should be empty. Check response example.

Tables

========================

CREATE TABLE `antz\_medical\_tests` (

`id` int unsigned NOT NULL AUTO\_INCREMENT,

`lab\_test\_id` text,

`is\_additional\_sample` tinyint DEFAULT '0',

`additional\_samples` text,

`medical\_record\_id` int NOT NULL,

`animal\_id` int unsigned NOT NULL,

`sample\_id` int unsigned DEFAULT NULL,

`sample\_name` varchar(100) DEFAULT NULL,

`test\_id` int unsigned DEFAULT NULL,

`test\_name` varchar(100) DEFAULT NULL,

`parent\_test\_id` int unsigned DEFAULT NULL,

`parent\_test\_name` varchar(100) CHARACTER SET utf8mb3 COLLATE utf8mb3\_general\_ci DEFAULT NULL,

`full\_test` tinyint NOT NULL DEFAULT '0' COMMENT '0 = No, 1 = Yes',

`is\_deleted` tinyint NOT NULL DEFAULT '0' COMMENT '0 = No, 1 = Yes',

`created\_by` int NOT NULL,

`modified\_by` int NOT NULL,

`created\_at` datetime DEFAULT CURRENT\_TIMESTAMP,

`modified\_at` datetime DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

`lab\_id` int unsigned DEFAULT NULL,

`site\_id` int unsigned DEFAULT NULL,

`status` enum('pending','inprogress','completed','transferred') DEFAULT 'pending',

`transfer\_reason` text,

`is\_tranferred` tinyint DEFAULT '0',

PRIMARY KEY (`id`),

KEY `fk\_antz\_medical\_tests\_medical\_record\_id` (`medical\_record\_id`),

KEY `fk\_antz\_medical\_tests\_animal\_id` (`animal\_id`),

KEY `fk\_antz\_medical\_tests\_sample\_id` (`sample\_id`),

KEY `fk\_antz\_medical\_tests\_test\_id` (`test\_id`),

KEY `fk\_antz\_medical\_tests\_parent\_test\_id` (`parent\_test\_id`),

CONSTRAINT `fk\_antz\_medical\_tests\_animal\_id` FOREIGN KEY (`animal\_id`) REFERENCES `antz\_animals` (`animal\_id`),

CONSTRAINT `fk\_antz\_medical\_tests\_medical\_record\_id` FOREIGN KEY (`medical\_record\_id`) REFERENCES `antz\_medical\_record` (`id`),

CONSTRAINT `fk\_antz\_medical\_tests\_parent\_test\_id` FOREIGN KEY (`parent\_test\_id`) REFERENCES `antz\_master\_lab\_tests` (`id`),

CONSTRAINT `fk\_antz\_medical\_tests\_sample\_id` FOREIGN KEY (`sample\_id`) REFERENCES `antz\_master\_samples\_types` (`id`),

CONSTRAINT `fk\_antz\_medical\_tests\_test\_id` FOREIGN KEY (`test\_id`) REFERENCES `antz\_master\_lab\_tests` (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=14115 DEFAULT CHARSET=utf8mb3;

CREATE TABLE `antz\_master\_samples\_types` (

`id` int unsigned NOT NULL AUTO\_INCREMENT,

`label` varchar(100) NOT NULL,

`string\_id` varchar(100) NOT NULL,

`description` text NOT NULL,

`zoo\_id` int DEFAULT '0',

`active` tinyint NOT NULL DEFAULT '1',

`created\_by` int NOT NULL,

`created\_at` datetime DEFAULT CURRENT\_TIMESTAMP,

PRIMARY KEY (`id`),

KEY `fk\_zoo\_id\_antz\_master\_samples\_types` (`zoo\_id`),

CONSTRAINT `fk\_zoo\_id\_antz\_master\_samples\_types` FOREIGN KEY (`zoo\_id`) REFERENCES `antz\_zoos` (`zoo\_id`) ON DELETE CASCADE

) ENGINE=InnoDB AUTO\_INCREMENT=17 DEFAULT CHARSET=utf8mb3;

CREATE TABLE `antz\_sample\_test\_mapping` (

`id` int unsigned NOT NULL AUTO\_INCREMENT,

`sample\_id` int unsigned NOT NULL,

`test\_id` int unsigned DEFAULT NULL,

`created\_by` int NOT NULL,

`modified\_by` int NOT NULL,

`created\_at` datetime DEFAULT CURRENT\_TIMESTAMP,

`modified\_at` datetime DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

PRIMARY KEY (`id`),

KEY `fk\_antz\_master\_samples\_types\_sample\_id` (`sample\_id`),

KEY `fk\_antz\_master\_lab\_tests\_test\_id` (`test\_id`),

CONSTRAINT `fk\_antz\_master\_lab\_tests\_test\_id` FOREIGN KEY (`test\_id`) REFERENCES `antz\_master\_lab\_tests` (`id`),

CONSTRAINT `fk\_antz\_master\_samples\_types\_sample\_id` FOREIGN KEY (`sample\_id`) REFERENCES `antz\_master\_samples\_types` (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=288 DEFAULT CHARSET=utf8mb3;

response example:

{

"success": true,

"data": [

{

"sample\_id": "1",

"sample\_name": "Whole Blood",

"tests": [

{

"test\_id": 1,

"test\_name": "Toxicology Tests",

"child\_tests": [

{

"test\_id": 2,

"test\_name": "Heavy Metal Testing",

},

{

"test\_id": 3,

"test\_name": "Drug Screening",

},

{

"test\_id": 4,

"test\_name": "Pesticide Residue Analysis",

}

],

},

{

"test\_id": 19,

"test\_name": "Renal Function Tests",

"child\_tests": [

{

"test\_id": 20,

"test\_name": "Glomerular Filtration Rate (GFR) Measurement",

}

],

},

{

"test\_id": 589,

"test\_name": "Microbiological Tests 2",

"child\_tests": [],

},

{

"test\_id": 590,

"test\_name": "Microbiological Tests 1",

"child\_tests": [],

}

]

},

{

"sample\_id": "16",

"sample\_name": "Blood spot, Egg Shells and Feathers",

"tests": [

{

"test\_id": 639,

"test\_name": "Sexing",

"child\_tests": [],

},

{

"test\_id": 640,

"test\_name": "Species Identification",

"child\_tests": [],

}

]

}

],

}