

SQL REPORTS

QUERY-1 (BONUS REPORT)

Business Requirement:

Time has come for the HR Manager of BlackChip semiconductors to roll-out the annual bonus for every employee. He has decided to set around 3% on overall profit generated by the projects worked by the employee as a bonus.

Description:

For every employee in every department, based on the number of orders he has worked on, we calculated the total profit generated across the orders. Further we computed the annual bonus of an employee by considering 3% of total profit generated by the projects worked by the employee as bonus.

Query used:

SELECT * FROM

(

SELECT tab3.employee_id,e.EMPLOYEE_FIRST_NAME,e.EMPLOYEE_LAST_NAME, e.Department_name,
COUNT(*) **AS** "Number of projects",

CONCAT('\$',round(SUM((s.Profit_percentage*tab3.budget)/100),2)) **AS** "Net profit",

concat('\$',round((SUM((s.Profit_percentage*tab3.budget)/100)*0.03),2)) **AS** Annual_bonus **FROM**

(

SELECT o.order_id, d.employee_id, o.budget **FROM** orders o **INNER JOIN** design d **ON**
o.order_id=d.order_id

UNION all

SELECT o.order_id, l.employee_id, o.budget **FROM** orders o **INNER JOIN** layout l **ON**
o.order_id=l.order_id

UNION all

SELECT o.order_id, v.employee_id, o.budget **FROM** orders o **INNER JOIN** verification v **ON**
o.order_id=v.order_id

UNION all

SELECT o.order_id, s.employee_id, o.budget **FROM** orders o **INNER JOIN** sales s **ON** o.order_id =
s.order_id) **AS** tab3

JOIN

sales s **ON** s.order_id=tab3.order_id

JOIN

SQL REPORTS

employees e **ON** tab3.employee_id=e.employee_id

GROUP BY tab3.employee_id **ORDER BY** Annual_Bonus DESC) **AS** tab33;

Functions used:

GROUP BY, ORDER BY, UNION ALL, JOIN, SUBQUERIES, SUM, CONCAT, ROUND.

Sample Report:

employee_id	EMPLOYEE_FIRST_NAME	EMPLOYEE_LAST_NAME	Department_name	Number of projects	Net profit	Annual_bonus
E13	vishwa	srivastava	Layout	1	\$198400.00	\$5952.00
E4	Chanikya	digumarty	Design	1	\$198400.00	\$5952.00
E1	Sai	bysani	Design	2	\$192500.00	\$5775.00
E10	Chethana	bohini	Layout	2	\$192500.00	\$5775.00
E18	kiana	reeves	Sales	3	\$1238500.00	\$37155.00
E8	Susmita	kancharla	Verif	3	\$1238500.00	\$37155.00
E11	janu	adabala	Layout	1	\$111600.00	\$3348.00
E2	Jahnvi	gamalapati	Design	1	\$111600.00	\$3348.00
E16	adwaith	menon	Layout	1	\$109200.00	\$3276.00
E6	Parinay	rikhy	Design	1	\$109200.00	\$3276.00
E5	Kavya	peta	Design	2	\$906000.00	\$27180.00
E15	nanditha	menon	Layout	1	\$900000.00	\$27000.00
E12	hari	krishna	Layout	3	\$766200.00	\$22986.00
E3	Praneetha	vallivedu	Design	3	\$766200.00	\$22986.00
E14	kartiki	bharghava	Layout	1	\$6000.00	\$180.00
E19	Di	Zhao	Sales	3	\$581400.00	\$17442.00
E9	Varshitha	nallu	Verif	3	\$581400.00	\$17442.00
E17	shannu	shabaj	Sales	4	\$464000.00	\$13920.00
E7	Rabbhani	sheik	Verif	4	\$464000.00	\$13920.00

SQL REPORTS

Query-2 (DESIGN TRACKING REPORT)

Business Requirement:

Blackchip semiconductors company wants to track their hardware designs, specifications, design cycle time, bugs resolving time as well as layout specifications in order to optimize their time and scope.

Description:

Based on the design start date and design end date, the design duration for each design has been determined. This will be utilized when the organization receives a new project and needs to pick which design team to assign it to finish it on time.

We've also computed the duration for each bug to be resolved, using the bugs logged in date and bugs logged out date as inputs. This time frame will assist the organization in determining how long it takes to fix a bug. The company will be able to estimate the time it will take to remedy the problem and the root cause of the problem by comparing it to the bugs description.

It also depends on the metal layer utilized for the chip and the voltage domain of the device in most circumstances. The company will get an idea about the faults that happened in a certain range by dividing the voltage domain into various ranges and comparing it to the bugs description.

Query :

```
SELECT d.design_id,DATEDIFF(d.dend_date,d.dstart_date) AS 'design_duration (days)',  
features,DATEDIFF(v.Bugs_logged_out_date,v.Bugs_logged_in_date) AS 'bug_resolving_duration  
(days)',  
v.bugs_description, l.layout_plan_description, l.metal_layer, l.voltage_domain FROM design d  
JOIN layout l ON d.design_id=l.design_id  
JOIN verification v ON l.design_id=v.design_id;
```

Functions used:

JOINS, DATEDIFF

SQL REPORTS**SAMPLE REPORT:**

	design_id	design_duration (days)	features	bug_resolving_duration (days)	bugs_Description	layout_plan_description	metal_layer	voltage_dom
▶	D1	31	80% DOU high frequency	11	Response time not within <1ns	Square	M1	0.675
	D10	17	low frequency 8nm support	1	Technode fail at 1ns	Wide square	G2	0.465
	D2	122	High frequency, low area	32	Trigger missing after 1 full cycle	Tabloid	M2	0.8
	D3	122	low power	32	Missing HTOL functionality	rectangle	M1	0.5
	D4	30	100% DOU savings	8	Fall response time is not within 4ns	wedge	M7	0.01
	D5	92	midrange support	19	Rise time drop in MX module	T-shape	M9	0.78
	D6	31	turbox support	10	chicken bit addition at 2nd bit	Narrow pipe	M10	0.675
	D7	14	turbo max support	4	Spare bit addition needed at core module	square	M0	0.655
	D8	123	low technode area	23	High frequency usecase implementation	rectangle	D3	0.515
	D9	153	low frequency support	18	Low power failure due to 1b2b connection	T-shape	G1	0.615

Query-3 (CLIENT SUMMARY REPORT)**Business Requirement:**

The sales team of Blackchip semiconductors company is given a task of finding high profitable clients among the existing ones.

Description:

In our database, we have four clients belonging to either one or more locations. In our report, we reported profit that is generated by every client, this profit information is obtained from the sales table, since there are multiple locations from a single client, we also found the total number of locations for each client.

Finding this is critical for the company to prioritize and plan their future client assignments and deliverables.

Query:

SELECT client_id,

IF(client_id='C1','Dell',if(client_id='C2','Toshiba',if(client_id='C3','Seagate','Western Digital')))) **AS**
client_name,

no_of_locations,

no_of_orders,SUM_of_order_size,

Clients_total_budget_\$,profit_\$

FROM (

SELECT o.client_id,**COUNT**(o.location_id) **AS** no_of_locations,

COUNT(o.order_id) **AS** no_of_orders,

SQL REPORTS

```
SUM(o.order_size) AS SUM_of_order_size,  
round(SUM(o.budget),2) AS Clients_total_budget_$,  
round(SUM((s.budget*s.profit_percentage)/100),2) AS profit_$  
FROM sales s  
JOIN orders o ON s.order_id=o.order_id  
GROUP BY o.client_id ORDER BY profit_$ DESC  
) AS t2 ;
```

Functions used:

SUBQUERY, CONCAT, GROUP BY, ORDER BY, SUM, JOINS

Sample Report:

client_id	client_name	no_of_locations	no_of_orders	SUM_of_order_size	Clients_total_budget_\$	profit_\$
C4	Western Digital	2	2	51500	2020000	1009200.00
C2	Toshiba	3	3	23500	2120000	670600.00
C3	Seagate	2	2	16000	1030000	354000.00
C1	Dell	3	3	25000	2560000	304100.00

Query-4 (DEPARTMENT STATISTICS REPORT)

SQL REPORTS

Business Requirement:

The HR manager of Blackchip semiconductors wants to keep track of resources and their salary allocation across the several verticals of the company to make more informed decisions when hiring and allocating new resources to a team.

DESCRIPTION:

The number of employees in the design, layout, sales, and verification departments has been calculated.

Each employee's salary has been computed, and the salaries of each department's employees have been included. Here salary is computed by multiplying hourly rate, number of hours worked per day and number of days in a month (20 working days).

For each department, summary statistics such as the sum, average, minimum, and maximum salaries have been determined.

Various of these will help the organization understand the differences in pay across all departments. They can determine whether to increase or decrease their salary based on these statistics and the department's performance. The number of employees in each department will be useful in determining if the department requires additional or less staff to perform their assigned tasks.

SOLUTION 1:

```
SELECT department_name,IF(department_name='Design',COUNT(employee_id),  
IF(department_name='Verif',COUNT(employee_id),  
IF(department_name='Layout',COUNT(employee_id),  
IF(department_name='Sales',COUNT(employee_id),0)))) AS Total_no_employees,
```

```
IF(department_name='Design',SUM(hrs_worked_per_day*20*hourly_rate),  
IF(department_name='Verif',SUM(hrs_worked_per_day*20*hourly_rate),  
IF(department_name='Layout',SUM(hrs_worked_per_day*20*hourly_rate),  
IF(department_name='Sales',SUM(hrs_worked_per_day*20*hourly_rate),0)))) AS Total_Salary,
```

```
IF(department_name='Design',AVG(hrs_worked_per_day*20*hourly_rate),  
IF(department_name='Verif',AVG(hrs_worked_per_day*20*hourly_rate),  
IF(department_name='Layout',AVG(hrs_worked_per_day*20*hourly_rate),
```

SQL REPORTS

IF(department_name='Sales',**AVG**(hrs_worked_per_day*20*hourly_rate),0))) AS AVG_Salary,

IF(department_name='Design',**MAX**(hrs_worked_per_day*20*hourly_rate),

IF(department_name='Verif',**MAX**(hrs_worked_per_day*20*hourly_rate),

IF(department_name='Layout',**MAX**(hrs_worked_per_day*20*hourly_rate),

IF(department_name='Sales',**MAX**(hrs_worked_per_day*20*hourly_rate),0))) AS MAX_Salary,

IF(department_name='Design',**MIN**(hrs_worked_per_day*20*hourly_rate),

IF(department_name='Verif',**MIN**(hrs_worked_per_day*20*hourly_rate),

IF(department_name='Layout',**MIN**(hrs_worked_per_day*20*hourly_rate),

IF(department_name='Sales',**MIN**(hrs_worked_per_day*20*hourly_rate),0))) AS MIN_salary

FROM employees **GROUP BY** department_name;

SOLUTION 2:

SELECT department_name,**COUNT**(employee_id) **AS** NUMBER_OF_EMPLOYEES,

SUM(hrs_worked_per_day*20*hourly_rate) **AS** SUM_OF_SALARIES,

MAX(hrs_worked_per_day*20*hourly_rate) **AS** MAXIMUM_OF_SALARIES,

MIN(hrs_worked_per_day*20*hourly_rate) **AS** MINIMUM_OF_SALARIES,

ROUND(**AVG**(hrs_worked_per_day*20*hourly_rate),2) **AS** AVERAGE_OF_SALARIES

FROM employees **GROUP BY** department_name;

FUNCTIONS USED:

COUNT, SUM, MIN, MAX, AVG, GROUP BY

SAMPLE REPORT:

SQL REPORTS

department_name	number_of_employees	sum_of_salaries	maximum_of_salaries	minimum_of_salaries	average_of_salaries
Design	6	\$38400	\$7200	\$5600	\$6400.00
Layout	7	\$44800	\$7200	\$4800	\$6400.00
Sales	3	\$15200	\$5600	\$4800	\$5066.67
Verif	3	\$19200	\$7200	\$5600	\$6400.00

QUERY-5 (REGION WISE PROFIT SUMMARY REPORT)**Business Requirement:**

During the pandemic, the sales and profits have gone drastically down, the sales team of Blackchip semiconductors company is given a task of finding high profitable locations among the existing ones to prioritize and plan their future investments.

Description:

In our database, we have six locations from where one or more clients place orders. In our report, we reported profit that is generated from every location, this profit information is obtained from the sales table, since there are multiple clients from a single location, we also found the total number of clients per each location.

Further we used the following rule to classify the locations as high market, medium market, low market zone. This information is critical to company to prioritize and plan their future investments.

If profit < \$100K “Low market”, If profit > 500K “High market”, otherwise it is “Medium market”

Query used:**SELECT**

l.location_id, l.location_name, count(o.client_id) as no_of_clients, SUM(o.order_size) **AS** order_size,

CONCAT('\$',round(SUM((s.budget*s.profit_percentage)/100),2)) **AS** Profit,

CONCAT('\$',round((SUM((s.budget*s.profit_percentage)/100))/count(o.client_id),2)) **AS**

Normalized_profit_per_order ,

IF(round((SUM((s.budget*s.profit_percentage)/100))/count(o.client_id),2) < 100000,

' Low', **IF**(round((SUM((s.budget*s.profit_percentage)/100))/count(o.client_id),2) > 500000 ,

'High', 'Medium')) **AS** category from locations l

JOIN orders o **ON** l.location_id=o.location_id

SQL REPORTS

JOIN sales s **ON** s.order_id=o.order_id **GROUP BY** l.location_id **ORDER BY** normalized_profit_per_order **DESC**;

Functions used:

JOINS, IF, CONCAT, ORDER BY, GROUP BY

Sample Report:

location_id	location_name	no_of_clients	total_orders	profit	Normalized_profit_per_client	category
L5	rajasthan	1	50000	\$900000.00	\$900000.00	High
L4	mumbai	1	1000	\$60000.00	\$60000.00	Low
L2	bengaluru	1	10000	\$44500.00	\$44500.00	Low
L1	hyderabad	4	35500	\$982600.00	\$245650.00	Medium
L3	delhi	2	18000	\$241600.00	\$120800.00	Medium
L6	chennai	1	1500	\$109200.00	\$109200.00	Medium