

Business Reports:

Business Report Number One

city ▾	average_rating ▾	total_feedbacks ▾
Chicago	3	1
Las Vegas	3.5	2
Portland	4	2
Arlen	5	1

Narrative:

This query joins the customer and customer_feedback tables to link reviews customers provide with their respective locations. It then calculates the average rating per city to determine what areas have high and low satisfaction rates while counting the number of feedback entries per city to ensure reliability. Additionally, the query orders the ratings in ascending order so managers can quickly see which cities have room for improvement and focus on those first. BTS managers can understand which areas have low average ratings, which indicates customer dissatisfaction, and create plans for immediate service improvement. This query helps managers prioritize areas that need customer service efforts, and can offer ways to improve their services.

Query:

```
SELECT:
    customer.city,
    AVG(customer_feedback.rating) AS average_rating,
    COUNT(customer_feedback.feedback_id) AS total_feedbacks
FROM customer
INNER JOIN customer_feedback ON customer.customer_id = customer_feedback.customer_id
GROUP BY customer.city
ORDER BY average_rating ASC;
```

Business Report Number Two

Employee ID ▾	Employee ▾	Working Hours ▾	Total Workload ▾	Remaining Available Hour ▾
E196	Clara	40 7		33
E243	Sophie	35 0		35
E352	Dave	20 0		20
E683	Lauren	40 6		34
E971	Matt	40 1		39

Narrative:

This query is run on a weekly basis to establish who is most available and what the current workload is for each employee. The manager will perform this analysis as an assessment to aid in dividing up tasks and deciding who can take on new projects or urgent issues that need to be addressed. This is also a good way to gauge and analyze the busy season of the year once a year has passed and the data can be gathered all together. The report gathers employee ids, the employee name, the working hours they have, their total workload at the time of the query, and a calculation to display the remaining hours a particular worker has available in a forty hour work week. This is useful for dividing tasks, assessing employee workload, and ensuring that no one is falling behind, being left out, or being allowed to be too active or underactive in the workplace

Query:

```
SELECT e.employee_id AS [Employee ID], e.f_name AS Employee, e.work_hours_per_week
AS [Working Hours], NZ(SUM(et.hours_allocated_per_week), 0) AS [Total Workload],
(e.work_hours_per_week - NZ(SUM(et.hours_allocated_per_week), 0)) AS [Remaining
Available Hours]
FROM employee AS e LEFT JOIN (employee_task AS et LEFT JOIN project_task AS pt ON
et.task_id = pt.task_id) ON e.employee_id = et.employee_id
WHERE et.completion_date IS NULL
GROUP BY e.employee_id, e.f_name, e.work_hours_per_week;
```

Business Report Number Three

Employee ID ▾	Employee ▾	Office ID ▾	Skill ▾	Proficiency Level ▾
E196	Clara	O364	Python	Intermediate
E243	Sophie	O055	Python	Expert
E273	John	O282	Python	Beginner
E683	Lauren	O282	Python	Intermediate

Narrative:

This query is useful to analyze which employees may be useful for particular tasks or projects. A manager or project manager will know which task will be needed and if they do not know it will be accessible in the database. Once the required tasks are known, a manager can use the above workload table as well as this skill table to assess who may be best for the job. This query can also be edited by changing the skill name to search for a particular skill like SQL or Project Management. This is useful to assign employees to the proper task, assess where employees have high skill levels or where employees may need additional training or assistance, decide who may be best for a job that is of a certain caliber based not only on the fact that they have this skill, but that they also have a certain proficiency level. Also, we have added the office ID so that we can know where this employee is and which locations may have a high volume of employees with a certain skill.

Query:

```
SELECT e.employee_id AS [Employee ID], e.f_name AS Employee, e.office_id AS [Office ID],
s.skill_name AS [Skill], es.proficiency_level AS [Proficiency Level]
FROM (employee AS e
      INNER JOIN
      (SELECT * FROM employee_skill) AS es
      ON e.employee_id = es.employee_id)
INNER JOIN
(SELECT * FROM skill) AS s
ON es.skill_id = s.skill_id
WHERE s.skill_name = 'Python';
```

Business Report Number Four

company_name	city	state	rating	feedback_date	project_name	f_name	l_name	task_description	CountOfproject_id (PK)
CricketUnlimited	Las Vegas	Nevada	3	6/1/2009	cricket_web_design	Matt	Davis	DB Quality Assurance	1
DeeBirdLLC	Chicago	Illinois	3	9/1/2009	deebird_reorganization	Clara	Williams	Customer Updates	1
Markus&Sons	Las Vegas	Nevada	4	8/2/2009	markus_rebrand	Matt	Davis	DB Quality Assurance	1
ReynoldsLLC	Portland	Oregon	4	4/12/2009	reynolds_customer_identification	Clara	Williams	Statistical Analysis	2
ReynoldsLLC	Portland	Oregon	4	4/12/2009	reynolds_customer_identification	John	Smith	DB Design	2
ReynoldsLLC	Portland	Oregon	4	4/20/2009	reynolds_rebrand	Clara	Williams	Customer Updates	2
StricklandLLC	Arlen	Texas	5	4/13/2009	strickland_web_design	John	Smith	DB Design	1

Narrative:

This is our Customer Feedback Report. This shows our client their customers, the specific project that there is feedback on, the rating each customer gave as feedback (out of 5), the employee/s that worked on the project and the task they performed for the project, and lastly how many projects each client has given us. This query can be run on a weekly, monthly, or quarterly basis, depending on how often management would like to check into customer feedback. This can also be used during employee evaluations to determine who needs extra support, especially if used in conjunction with the previous query to determine hour availability and workload. This query was the result of combining two different queries into one result. The first query created a table with everything you see above except for “Countofprojects_id (PK)”

Three queries were written to generate the final Customer Feedback Report.

Query 1:

```
SELECT
customer.company_name, Count(project.[project_id (PK)]) AS [CountOfproject_id (PK)]
FROM
(customer INNER JOIN project ON customer.customer_id = project.[customer_id (FK)])
INNER JOIN customer_feedback ON (customer.customer_id = customer_feedback.customer_id)
AND (project.[project_id (PK)] = customer_feedback.project_id)
GROUP BY
Customer.company_name;
```

Result:

company_name	city	state	rating	feedback_date	project_name	f_name	l_name	task_description
CricketUnlimited	Las Vegas	Nevada	3	6/1/2009	cricket_web_design	Matt	Davis	DB Quality Assurance
DeeBirdLLC	Chicago	Illinois	3	9/1/2009	deebird_reorganization	Clara	Williams	Customer Updates
Markus&Sons	Las Vegas	Nevada	4	8/2/2009	markus_rebrand	Matt	Davis	DB Quality Assurance
ReynoldsLLC	Portland	Oregon	4	4/12/2009	reynolds_customer_identification	Clara	Williams	Statistical Analysis
ReynoldsLLC	Portland	Oregon	4	4/12/2009	reynolds_customer_identification	John	Smith	DB Design
ReynoldsLLC	Portland	Oregon	4	4/20/2009	reynolds_rebrand	Clara	Williams	Customer Updates
StricklandLLC	Arlen	Texas	5	4/13/2009	strickland_web_design	John	Smith	DB Design

This first query generates a report with the customer name, city and state they are located in, their feedback in the form of a score from 1 - 5, the date the feedback was submitted, the project name, the first and last name of the employee or employees assigned to the project, and the employees skill set.

Query 2:

```
SELECT
Customer_Feedback_Report1.company_name, Customer_Feedback_Report1.city,
Customer_Feedback_Report1.state, Customer_Feedback_Report1.rating,
Customer_Feedback_Report1.feedback_date, Customer_Feedback_Report1.project_name,
Customer_Feedback_Report1.f_name, Customer_Feedback_Report1.l_name,
Customer_Feedback_Report1.task_description, customer_feedback_report2.[CountOfproject_id
(PK)]
FROM
Customer_Feedback_Report1
INNER JOIN
customer_feedback_report2 ON Customer_Feedback_Report1.company_name =
customer_feedback_report2.company_name;
```

Result:

company_name	CountOfproject_id (PK)
CricketUnlimited	1
DeeBirdLLC	1
Markus&Sons	1
ReynoldsLLC	2
StricklandLLC	1

The second query generates a table with the customer name and how many projects they have contracted with Beaver Technology Solutions.

After results from both queries were done, they were combined to print the one report.

Final Query for Customer Feedback Report:

```
SELECT
customer.company_name, customer.city, customer.state, customer_feedback.rating,
customer_feedback.feedback_date, project.project_name, employee.f_name, employee.l_name,
project_task.task_description
FROM
((customer INNER JOIN customer_feedback ON customer.customer_id =
customer_feedback.customer_id)
INNER JOIN
project ON (project.[project_id (PK)] = customer_feedback.project_id) AND
(customer.customer_id = project.[customer_id (FK)]))
INNER JOIN
((project_task INNER JOIN (employee INNER JOIN employee_task ON employee.employee_id
= employee_task.employee_id) ON project_task.task_id = employee_task.task_id)
INNER JOIN task_project_association ON project_task.task_id =
task_project_association.task_id) ON project.[project_id (PK)] =
task_project_association.project_id
GROUP BY
customer.company_name, customer.city, customer.state, customer_feedback.rating,
customer_feedback.feedback_date, project.project_name, employee.f_name, employee.l_name,
project_task.task_description
ORDER BY
Customer.company_name;
```