If you have any questions or suggestions regarding the my feedback, please feel free to reach out to me on WhatsApp: **Sanjay Yadav Phone: 8310206130**

Hello, everyone! I wanted to share my recent experience from the SQL interview with **Abhishek Ringsia** from Mastercard, who has over 14 years of experience in the field.

Initially, Abhishek provided an agenda for the interview, mentioning that he would ask 4 questions - 3 queries and 1 schema. He explained the time allocation: the first 5 minutes for introductions, the next 20 minutes for queries, followed by 30 minutes for the schema, and the last 5 minutes for feedback.

After the introduction, we began with my journey and educational background from earning my bachelor's degree in 2003 to my current job. Abhishek presented 3 queries, and I efficiently answered them within the allocated time, confirming expectations and explaining my thought process. Here are the quries

Q1. Suppose two tables, Students and Courses exist. Students includes Student_ID, Name, and Course_ID, and Courses includes Course_ID, Course_Name, and Credits. Write a SQL query to list all courses along with the names of the students enrolled in them, ensuring to include courses that currently have no students.

Ans--

```
Students =======================        Courses============================
Student_ID, Name, Course_ID             Course_ID, Course_Name,Credits

SELECT c.Course_Name, s.Name
FROM Courses c
LEFT JOIN Students s ON c.Course_ID = s.Course_ID;
```

Q2. Suppose we have Orders and Customers tables. Orders includes Order_ID, Customer_ID, and Order_Amount, while Customers includes Customer_ID, Customer_Name, and City. Write a SQL query to find the cities which have customers who have placed more than 10 orders with a total Order_Amount of more than $10,000.

Ans--

```
Orders ==========================      Customers=======================
Order_ID, Customer_ID, Order_Amount     Customer_ID, Customer_Name, City

SELECT DISTINCT c.city
FROM Customers c
JOIN Orders o ON c.Customer_ID = o.Customer_ID
GROUP BY c.city
HAVING COUNT(c.Customer_ID) > 10
       AND SUM(o.Order_Amount) > $10,000;
```

Q3. Suppose we have Orders and Customers tables. Orders includes Order_ID, Customer_ID, and Order_Amount, and Customers includes Customer_ID, Customer_Name, and City. Write a SQL query to find customers who have made orders totaling more than the average order amount of all customers.

Ans--

Orders=============================== Customers ================================
Order_ID, Customer_ID, Order_Amount Customer_ID, Customer_Name, City

```
SELECT c.Customer_Name
FROM Customers c
JOIN Orders o ON c.Customer_ID = o.Customer_ID
GROUP BY c.Customer_ID, c.Customer_Name
HAVING SUM(Order_Amount) > (
    SELECT AVG(Order_Amount) FROM Orders
    );
```

Moving on to the schema, I discussed my approach, assumptions, and began creating entities and attributes. While constructing the schema, I continuously explained my actions and reasoning to Abhishek. Towards the end, I detailed the relationship tables and proposed a scalable approach. Abhishek agreed with my suggestions, and I completed the schema within the 30-minute requirement.

Q4. Healthcare and Insurance Management System Use Cases
- A system containing multiple Hospitals, each with a unique identifier, name, and location.
- Each Hospital has multiple Departments, and each Department can be staffed by multiple Doctors.
- Each Doctor can work in multiple hospitals and have multiple Patients assigned.
- Patients are registered in the system with their unique identifier, name, and insurance details.
- Each Patient's visit records to different Hospitals, their medical tests, prescriptions, and billing details are tracked.
- The insurance details, claims, and settlements for each Patient are also maintained in the system.

My solution ---

```
----------------------------------------------------------
Entity Tables
-----------------------------------------------------------
hospital ===============================
id (pk), name, location

department ===========================
id (pk), name, hospital_id (fk)

doctor ===============================
id (pk), name, specility

patient ==============================
id (pk), name

insurance_provider ====================
id (pk), name, coverage_type
```

Relation Tables
----
doctor_department ======================
(doctor_id, department_id) (pk)

doctor_patient ======================
(doctor_id, patient_id)(pk)

patient_insurance_detail ==================
id(pk), patient_id, insurance_provider_id, start_date, end_date

patient_record ========================
id (pk), patient_id, hospital_id, , test, prescription

hospital_inventray ======================
id, patient_id, hospital_id, detail

During the feedback session, Abhishek expressed his satisfaction, commending my correctness in queries, adherence to schema requirements, and my professional approach to scalability. Despite the positive feedback, when I checked the dashboard, I was surprised to see a score of 7 out of 10 with a weak hire status. This left me disheartened as I felt confident in my responses and the positive feedback received during the interview.

During the feedback session Abhishek said he was happy with my answers and how I handled the schema and my professional approach to scalability in the interview. But when I saw my score later, it was only 7 out of 10, and I was labeled as a weak hire. This was unexpected and made me feel disappointed because I thought I did well in the interview based on the positive feedback.

In an effort to seek clarification, I decided to email Abhishek directly. I located the email from the interview invitation and sent the following email:

==== Email sent by me ==

Hello Abhishek

Hello, this is Sanjay from Scaler February23. I completed my SQL interview with you at 7:30 pm on Nov 15. Thank you for providing me with the opportunity to showcase my SQL skills and for your feedback. Based on our feedback discussion, I was anticipating a strong positive evaluation from your side, given that all three queries were correct and you rated the schema 9 out of 10.

If you don't mind, may I ask for the reason behind the lower rating? I was expecting at least a 9 out of 10 with a strong recommendation, which would significantly benefit my master's degree course. I would be very grateful if you could reconsider and potentially revise the interview ratings if I am eligible for such a reconsideration.

After a while, I received a response from the instructor, and they raised a support ticket with Scaler to address the issue

==== Replay from interviwer ===

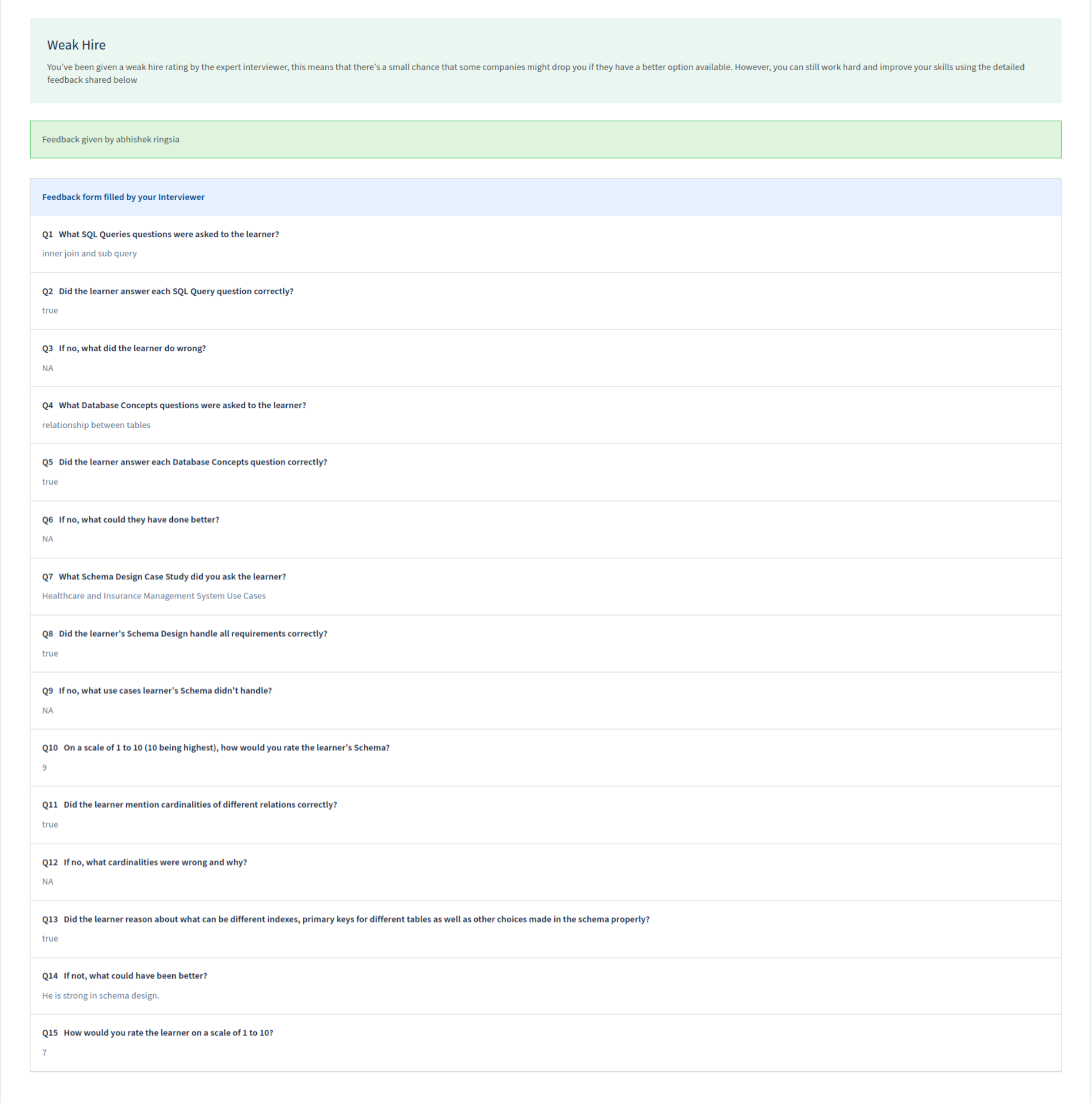
Hi Scaler support,

In today's interview with Sanjay, i mistakenly pressed 7 after giving 9 rating and form got submitted quickly. I tried to update the feedback but I don't see any option to do that. Can you please help us to update the form and change final rating to 9.

@sanjay - My intent was to fill 9 rating but due to network lag while filling form it got overwritten I guess.

===

When I reached out to the support team, they informed me that this issue is unique and involves backend intervention. The resolution time is uncertain, but your interviewer has elevated the matter by creating a support ticket, given its highest priority. I'm eagerly awaiting the update to rectify my grades. Below is the feedback provided by the interviewer.



Weak Hire

You've been given a weak hire rating by the expert interviewer, this means that there's a small chance that some companies might drop you if they have a better option available. However, you can still work hard and improve your skills using the detailed feedback shared below

Feedback given by abhishek ringsia

Feedback form filled by your Interviewer

Q1 What SQL Queries questions were asked to the learner?
inner join and sub query

Q2 Did the learner answer each SQL Query question correctly?
true

Q3 If no, what did the learner do wrong?
NA

Q4 What Database Concepts questions were asked to the learner?
relationship between tables

Q5 Did the learner answer each Database Concepts question correctly?
true

Q6 If no, what could they have done better?
NA

Q7 What Schema Design Case Study did you ask the learner?
Healthcare and Insurance Management System Use Cases

Q8 Did the learner's Schema Design handle all requirements correctly?
true

Q9 If no, what use cases learner's Schema didn't handle?
NA

Q10 On a scale of 1 to 10 (10 being highest), how would you rate the learner's Schema?
9

Q11 Did the learner mention cardinalities of different relations correctly?
true

Q12 If no, what cardinalities were wrong and why?
NA

Q13 Did the learner reason about what can be different indexes, primary keys for different tables as well as other choices made in the schema properly?
true

Q14 If not, what could have been better?
He is strong in schema design.

Q15 How would you rate the learner on a scale of 1 to 10?
7

Sql notes ->
https://drive.google.com/file/d/1ER1C2_4_pUwgDKDSKHp6pyTZR3JUu44r/view?usp=sharing