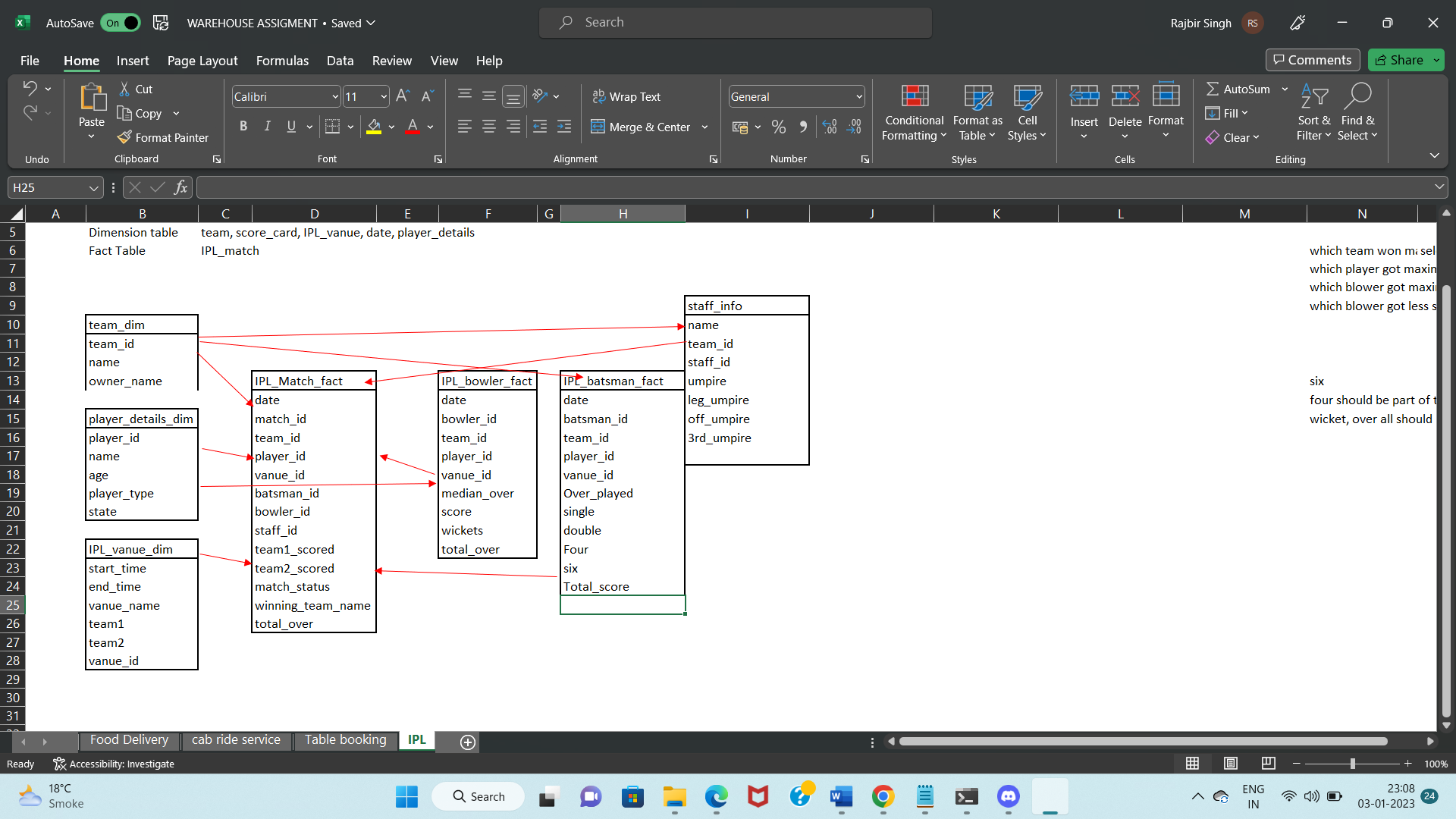
1. **Design a Data Warehouse for IPL Cricket Tournament (Asked in Flipkart Interview for Senior Data Engineer role)**

**Grains: -** each row for each match

Dimension table: - team, IPL\_vanue, player\_details, staff\_info

Fact table: - IPL\_match, IPL\_batsman, IPL\_bwoler

|  |  |
| --- | --- |
| Report | Query |
| which team won maximum matches | select count(winning\_team\_name) as win\_count from IPL\_Match order by win\_count desc; |
| which player got maximum score | select player.name, sum(bats.total\_score) as total\_score from player\_details\_dim player inner join ipl\_batsman\_fact bats on player.player\_id = bats.player\_id order by total\_score desc; |
| which blower got maximum wicket | select player.name, sum(ball.wickets) as total\_wickets from player\_details\_dim player inner join ipl\_bowler\_fact ball on player.player\_id = bats.player\_id order by total\_wickets desc; |



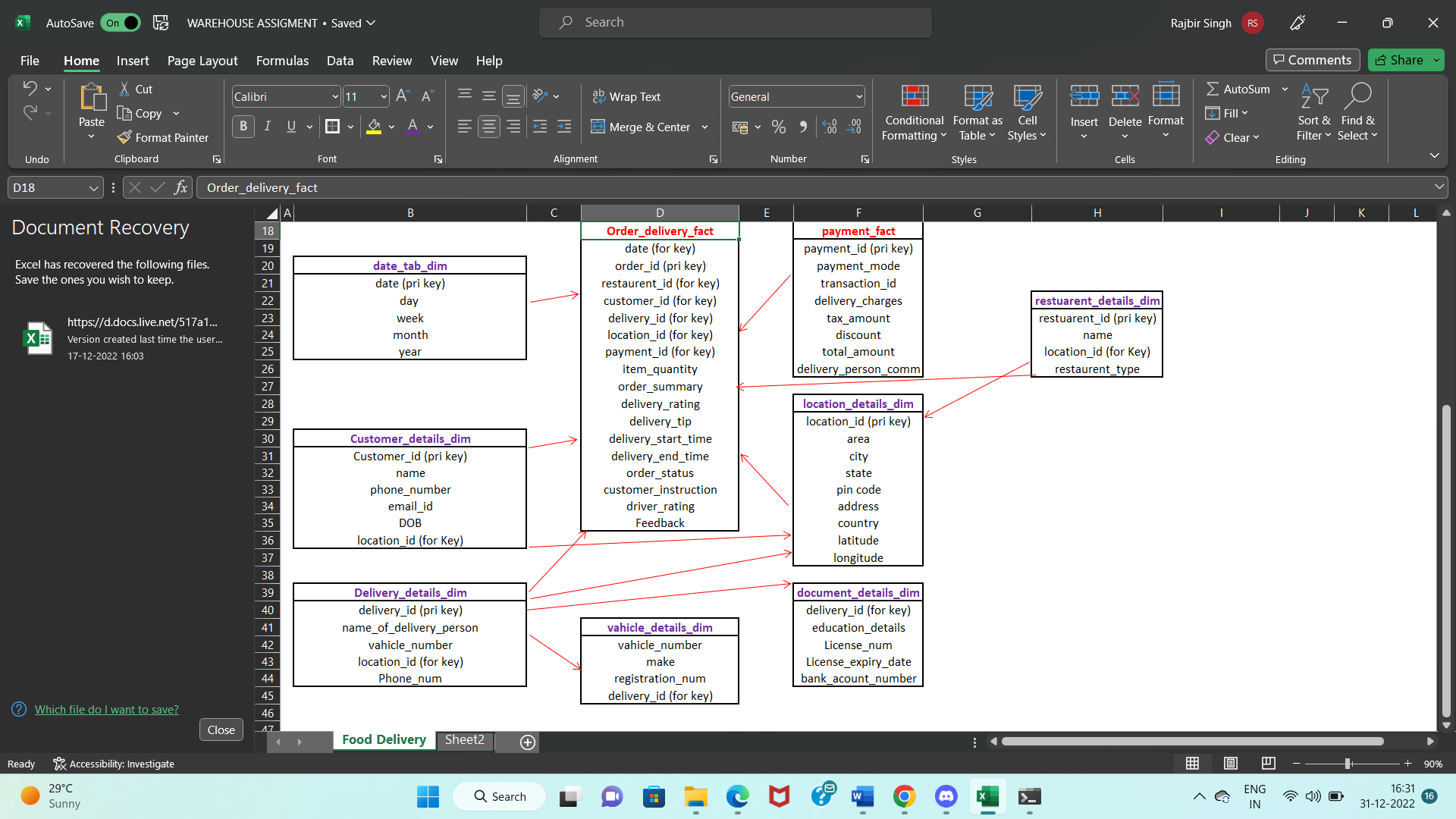
**2. Design a Data Warehouse for Food delivery app like Swiggy, Zomato (Asked in Grab for Data Engineer role)**

|  |  |  |
| --- | --- | --- |
| S.no | Analysis | Queries |
| 1 | How many orders delivered every day | Select date, count(\*) as order\_count from order\_delivery\_fact where order\_status = ‘delivered’ group by date; |
| 2 | from which city maximum order delivered for last 7 days | Select location.city, order.count(\*) as order\_count from order\_delivery\_fact order left join location\_details location on order.location\_id = location.location\_id and order.date between ‘2022-12-23’ and ‘2022-12-30’ group by location.city order by order\_count desc; |
| 3 | Total revenue every day | select order.date, sum(payment.total\_amount) as total\_sales from order\_delivery\_fact order left join payment\_fact payment on payment.payment\_id = order.payment\_id group by order.date; |
| 4 | Number of cancellations everyday | Select date, order\_status as cancellation\_count from order\_delivery\_fact where order\_status = ‘Cancel’ group by date; |

**Grains:** - Need to track order delivery transaction

**Dimension tables: -** Date, Customer, Restaurant, Delivery boy, Documents, Vahicle\_type, Location

**Fact tables: -** Order delivery & Payment



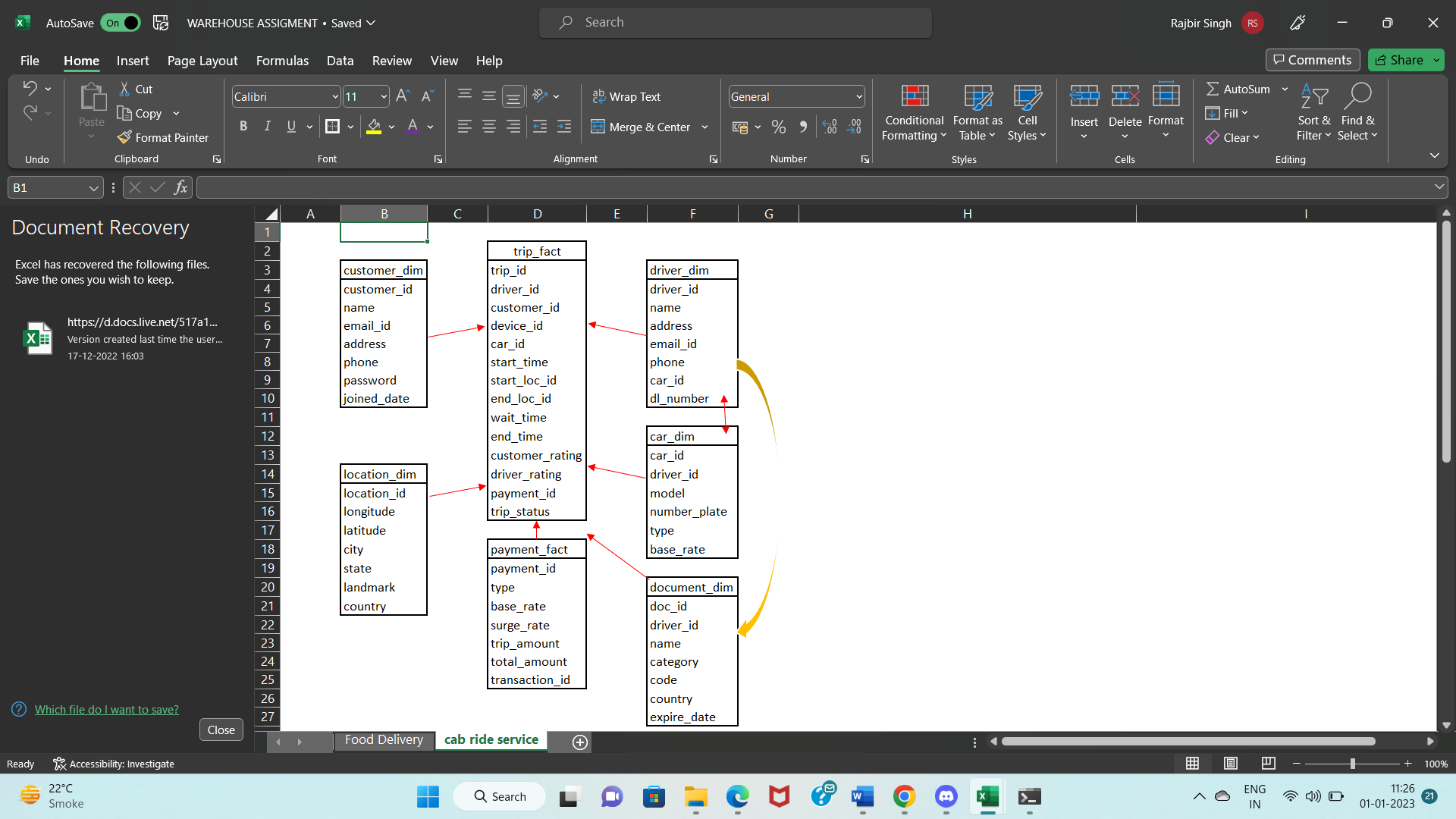
**3. Design a Data Warehouse for cab ride service like Uber, Lyft (Asked in Google for Data Engineer role)**

**Grains:** - need to track individual trip at each transaction

**Dimension tables: -** customer, location, driver, car, document

**Fact tables: -** trip & Payment

|  |  |
| --- | --- |
| **Reports** | **queries** |
| Track ride done by driver and their performance | select driver\_id, customer\_rating from trip\_fact where trip\_status = 'completed'; |
| how many rides are happening to a common or famous destination each day | select date, count(\*) from trip\_fact group by date; |
| How many trips cancelled each day | select start\_time, count(\*) from trip\_fact where trip\_status = 'cancelled' group by start\_time; |
| How many rides and the average price during the peak hours per day | select date\_format(trip.start\_time,'%d.%m.%Y') as date, count(trip.trip\_status) as rides\_count, avg(payment.total\_amount) as average\_price from trip\_fact trip inner join payment on trip.payment\_id =payment.payment\_id and time(trip.start\_time) between '18:00:00' and '21:00:00' group by date; |



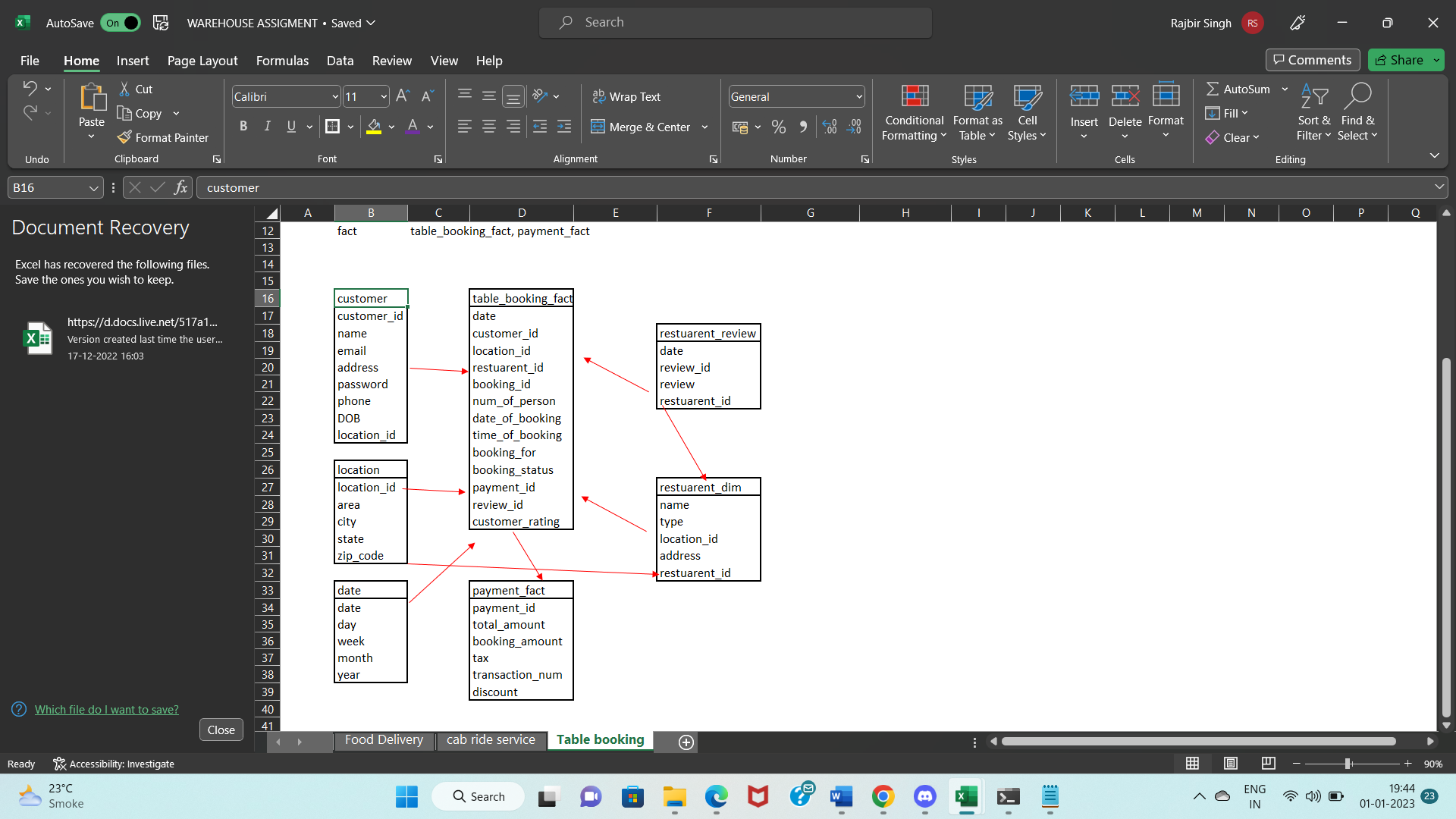
**4. Design a Data Warehouse for Restaurent table booking app like Dineout (Asked in McKinsey for Consultant Data Engineer role)**

Grains: - each row customer's restaurant booking info

Dimension table: - customer, location, restaurant, payment, date, review

Fact table: - table\_booking\_fact, payment\_fact

|  |  |
| --- | --- |
| Report | Queries |
| top 10 restaurant name | select res.name, avg(booking.customer\_rating) as rating from table\_booking\_fact booking inner join restaurant\_dim res on res.restaurant\_id = booking.restaurant\_id order by rating desc limit 10; |
| Lowest rating restaurant | select res.name, avg(booking.customer\_rating) as rating from table\_booking\_fact booking inner join restaurant\_dim res on res.restaurant\_id = booking.restaurant\_id order by rating ; |
| most frequent visit restaurant | select count(restaurant\_id) as visit\_count from table\_booking\_fact order by visit\_count desc; |



**5. Design a Data Warehouse for Covid Vaccination Application (Asked in Livsapce for Data Engineer role**

**Grains: -** person wise vaccination details in each row

**Dimension tables: -** date, customer, location, vaccine\_info\_dim, company\_info\_dim

**Fact table:** **-** covid\_vaccination\_application\_fact

|  |  |
| --- | --- |
| Analysis | Query |
| day wise vaccine count | select date, count(date) as vaccine\_count from covid\_vaccination\_application\_fact group by date; |
| which area has got maximum number of vaccine | select location.area, count(covid.date) as area\_count from covid\_vaccination\_application\_fact covid inner join location\_dim location on location.location\_id = covid.location\_id group by location.area; |
| proportion of 1st or 2nd dose | SELECT (count(1st\_dose)/count(date)) as 1st\_dose\_propostion, (count(2nd\_dose)/count(date)) as 2nd\_dose\_proportion from covid\_vaccination\_application\_fact; |

