MGMT 58200

Management of Organizational Data

Final Project

Group 1:

Ajayi, Oluwayemisi Adejoke

Chopra, Nikhil

Dalal, Mrinmoy

Kamat, Pratik Suhas

Satpathi, Anushila

Business Background and Overview

Ernie's BWC Auto Care is a popular auto-repair shop in 609 Sagamore Parkway North, Lafayette that specializes in car repair and reconditioning. In particular, they specialize in collision repair, exhaust, auto repair, suspension, tires, wheels, and custom work.

The auto shop is a family owned and run shop having about 18 employees (7 full time/ permanent and 11 contractual employees). The owner of the shop Mr. Lopez opened this shop single-handedly, 15 years ago, when he immigrated to Indiana all the way from Mexico along with his wife and two sons just old enough to call him 'Papá'. 15 years back, Mr. Lopez started the shop just as a tire changing shop and with the support of his wife and two sons has been able to grow the business to a full-fledged auto repair shop in Lafayette, Indiana.

As you might have guessed, Mr. Lopez is an extremely ambitious businessman and has big dreams for his auto-repair shop. Mr. Lopez plans on retiring in 2024 and before he retires, he aims to make his shop the biggest in Lafayette. For this he understands he must develop a double-edged growth strategy that leverages both technology and solving business related issues and concerns.

Current Business Scenario

The auto-repair shop has been performing extremely well since the past few years, especially since after COVID restrictions have been lifted. In the last financial year, the shop generated a revenue of around $1.5 Million, which is a fantastic figure. However, this figure is not which keeps Mr. Lopez up in the night, it is the amount he can take home as clean profit. The profit currently stands at less than 10% (8% to be specific) and thus even though the shop generates $1.5 million, Mr. Lopez and family are only able to take home USD 120,000 – which doesn't seem like a great figure.

With multiple conversations with Mr. Lopez in the attempt to understand the business process and operations involved, we were able to identify some of the key challenges impeding the shop's profitability and growth.

Business Related Challenges

- Resource Planning
- Inventory Management
- Order Planning
- HR Issues

Technology Related Challenges

Mr. Lopez currently doesn't have any sophisticated method to store, process and analyse the data his shop generates. He does not even have a sophisticated database design, and here we would be coming to his aid by utilizing our knowledge of database management to create a sophisticated database model for his shop and derive a data-based growth strategy.

Overview of the Project Objectives

Stage1: Database Design and Conceptual Data Modelling

- Understand each of the collected attributes and develop independent data tables based on our understanding of the business
- Develop a sophisticated Entity Relationship Diagram (ERD) based on the developed tabular data
- Create a relational schema based on the ERD and identify primary and foreign keys
- Execute a normalization analysis based on the above steps

Stage2: Database Implementation and SQL Querying

- Execute SQL script(s) to create the tables and test following your relational schema
- Perform interesting SQL queries to identify business opportunities

Stage3: Hypothesizing Business Recommendations and Insights

- Generate business related insights based on database design and implementation stages
- Collate actionable business recommendations aimed to improve profitability and overall functioning of the auto care shop

Stage 1: Database Design and Conceptual Data Modelling

Introduction to various entities being captured in the data and different tables

Data Table	Table Description
Job	Contains the unique job IDs, status of job (active/complete), start time, end time
Invoice	Information on unique invoice numbers, date, tax, discount, total amount
Customer	Information of each unique customer – name (first + last), demographics (address, email, phone number), customer type (business / individual)
Vehicle	Information of each unique vehicle – model, make, category
Service	List of unique services offered – service ID, description, service base rate
Employee	Employee information – name, demographics, type (full time / contract)
Consumable	List of consumables in the inventory – unit price, quantity, description
Job_consum	Associative relational table between job and consumable tables
Job_service	Associative relational table between job and service tables
Employee_job	Associative relational table between job and employee tables

Stage 1: Database Design and Conceptual Data Modelling

Data tables and corresponding primary and foreign keys

Data Table	Primary Key	Foreign Keys
Job	**job_id**	**Cust_vin, vin**
Invoice	Invoice_no	Job_id
Customer	Cust_id	-
Vehicle	Vin	-
Service	Emp_id	-
Employee	Consum_id	-
Consumable	Service_id	-
Job_consum	Job_id, consum_id	Job_id, consum_id
Job_service	Service_id, job_id	Service_id, job_id
Employee_job	Job_id, emp_id	Job_id, emp_id

** 2 additional tables are present (Customer_Phone and Employee_Phone) which are multi-valued tables and contain the phone numbers of customers and employees

*** 2 additional tables are specialized is a tables (Contract_Employee and FullTime_Employee)

Assumption and constraints for database design

1) No direct link between Vehicle and Customer, as they are being linked indirectly through the Job table.

2) Only documenting the visits that convert to a job

3) Not tracking quotations as a separate table, Service table to be used to compute base rates

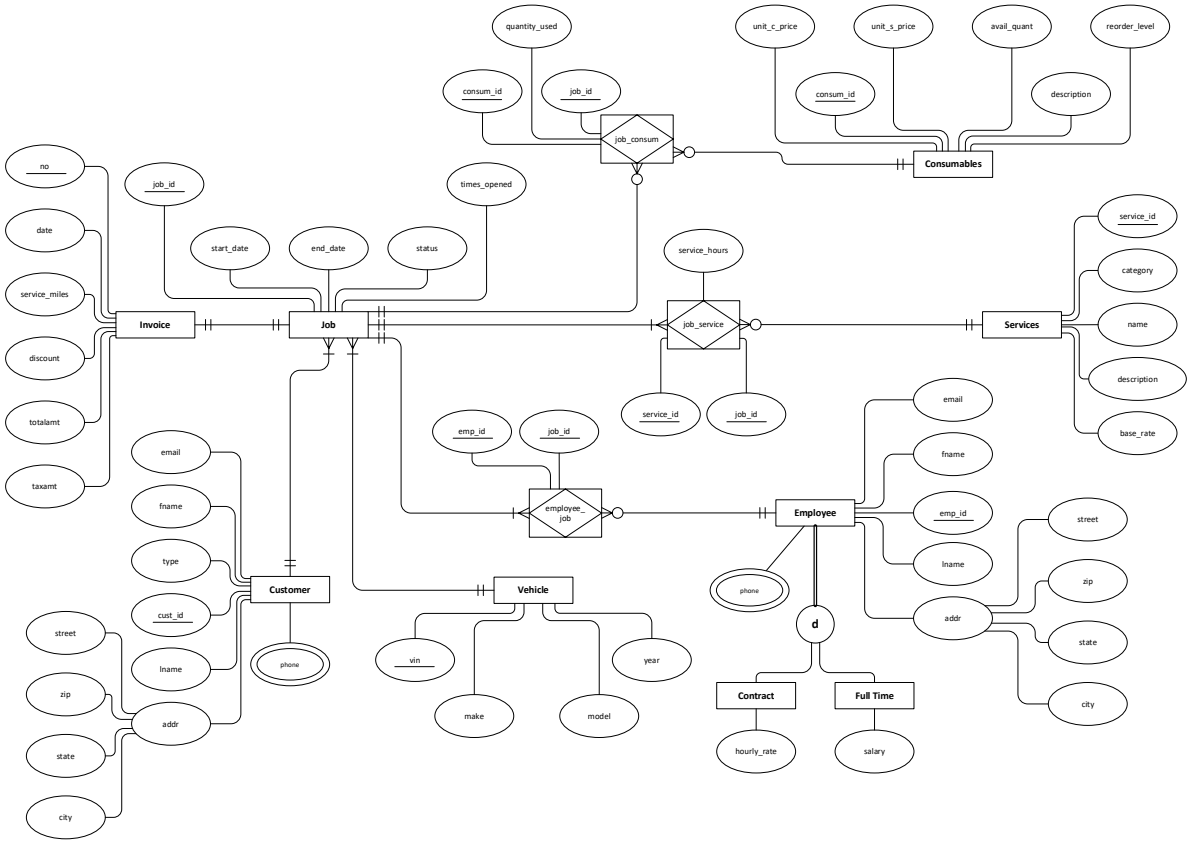
Stage 1: Database Design and Conceptual Data Modelling

Business Rules to create entity relationship diagram

1. Customer – Job:
    a. Customer can open one or more jobs (more than one car / family).
    b. One job can be opened by only one customer.
2. Invoice – Job:
    a. One job has one invoice.
    b. One invoice can be of only one job.
3. Vehicle – Job:
    a. One job is for one vehicle only (for multiple vehicles, multiple jobs will be opened for a customer).
    b. One vehicle may be associated with one or more jobs
4. Consumables – Job:
    a. One job may require zero or one consumables.
    b. One part may be used by one or more jobs
5. Service – Job:
    b. One job will require at least one service
    c. One service can be required for multiple different jobs.
6. Employee – Job:
    a. An employee may work on zero or many jobs (at once or during a time frame).
    b. A job needs at least one employee.

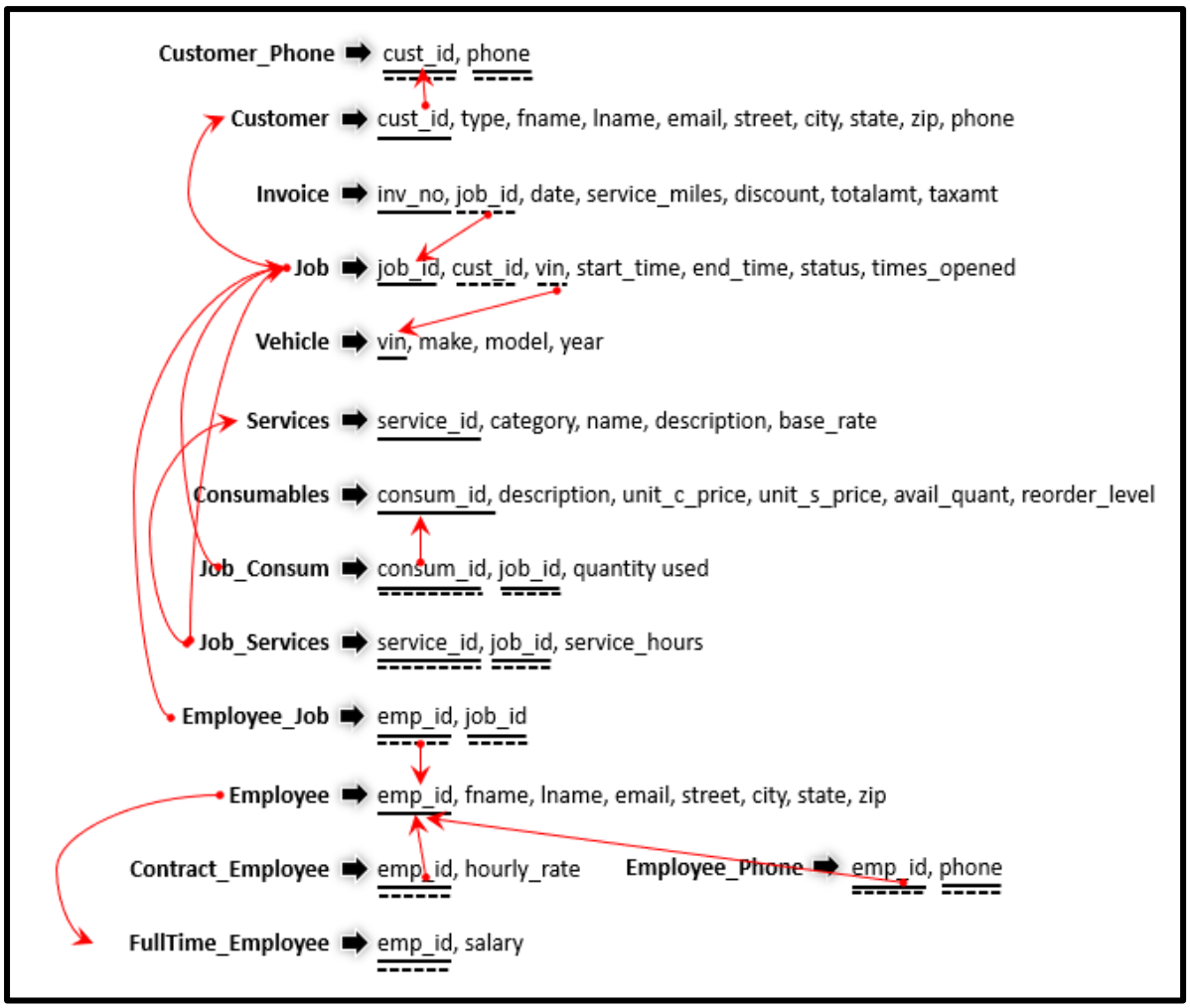
Stage 1: Database Design and Conceptual Data Modelling

Entity Relationship Diagram (ERD)



Stage 1: Database Design and Conceptual Data Modelling

Development of relational schema



Stage 1: Database Design and Conceptual Data Modelling

Normalization Analysis

Once the relational schema was prepared, it it was checked to ensure that there were no anomalies in inserting, deleting, or updating data with examples provided below:

Table 1: Relational Database with Anomalies

JOB_ID	Job_start_date	Service_Id	Service_Name
JB_ID10001	06/04/2022	S1_001	Paint(original)
JB_ID10002	06/09/2022	S1_001	Paint(original)
JB_ID10003	08/08/2022	S1_002	Bolt on Kits

Table 2: Relational Database with no Anomalies

JOB_ID	Job_start_date	Service_Id
JB_ID10001	06/04/2022	S1_001
JB_ID10002	06/09/2022	S1_001
JB_ID10003	08/08/2022	S1_002

Service_Id	Service_Name
S1_001	Paint(original)
S1_002	Bolt on Kits

- ***Insertion Anomaly:*** *Jobs and services are stored in separate tables instead of one table so that information on new services can be inserted into the service table even if no job is associated with it*
- ***Deletion Anomaly:*** *If one service was only associated with one job, deletion of the job ID would result in deletion of that service information*
- ***Update Anomaly:*** *If the service name was required to be updated, keeping a separate services table would ensure only one update is required instead of updating multiple rows*

Stage 1: Database Design and Conceptual Data Modelling

Normalization Analysis

Each table in the relational database was checked and was found to be in Third Normal Form (3NF)

Example of 1st Normal Form: Atomic Values in each table record

JOB_ID	Job_start_date	Job_end_date	Times_reopened	Job_status	VIN	Cust_id
JB_ID10001	06/04/2022	29/04/2022	0	Closed	19UYA31581L000006	C000001
JB_ID10002	06/09/2022		0	Active	YS3CC55B6N9000072	C000002

Example of 2nd Normal Form: Fully functionally dependent on primary key

Functional Dependencies:

Job_id → Job_StartDate, Job_EndDate, Times_reopened, Job_status, VIN, Cust_id

- **First Normal Form:** No table contains any multi-valued or composite attributes hence they are in First Normal Form
- **Second Normal Form**: Every table is in First Normal Form and each non-key attribute is fully functionally dependent on the primary key
- **Third Normal Form:** Every table is in Second Normal Form and no transitive dependency exists for non-key attributes

Stage 2: Database Implementation and SQL Querying

Setting Primary and Foreign Keys and Datatypes

```
set foreign_key_checks = 0;

#Customers Table - Change data type to varchar and add primary key
alter table customers modify column cust_id varchar(255);
alter table customers add primary key(cust_id);

#Customer Phone Table - Change datatype and add primary and foreign key constraints
alter table customer_phone modify column cust_id varchar(255), modify column phone varchar(255);
alter table customer_phone add primary key(cust_id, phone);
alter table customer_phone add foreign key (cust_id) references customers(cust_id);

#Employee Table - Change datatype and add primary key constraint
alter table employee modify column emp_id varchar(255);
alter table employee add primary key (emp_id);

#Employee_Phone Table - Change datatype and add key constraints
delete from employee_phone where emp_id = "emp_id";
alter table employee_phone modify column emp_id varchar(255), modify column phone varchar(255);
alter table employee_phone add primary key(emp_id, phone);
alter table employee_phone add foreign key (emp_id) references employee(emp_id);

#Consumables Table - Change datatype and add primary key constraint
alter table consumables modify column consum_id varchar(255), modify column unit_c_price double;
alter table consumables add primary key (consum_id);
```

```
#Contract_Employee Table - Change datatype and add primary key constraint
alter table contract_employee modify column emp_id varchar(255);
alter table contract_employee add primary key (emp_id);

#Fulltime_Employee Table - Change datatype and add primary key constraint
alter table fulltime_employee modify column emp_id varchar(255);
alter table fulltime_employee add primary key (emp_id);

#Service Table - Change datatype and add primary key constraint
alter table services modify column service_id varchar(255);
alter table services add primary key (service_id);

#Vehicle Table - Change datatype and add primary key constraint
alter table vehicle modify column vin varchar(255);
alter table vehicle add primary key (vin);

#Jobs Table - Update datetime, change datatype and add key constraints
update jobs set start_datetime = STR_TO_DATE(start_datetime, "%m/%d/%Y %H:%i:%s");
update jobs set end_datetime = STR_TO_DATE(end_datetime, "%m/%d/%Y %H:%i:%s");
alter table jobs modify column job_id varchar(255), modify column vin varchar(255), modify column cust_id varchar(255);
alter table jobs add primary key(job_id);
alter table jobs add foreign key(vin) references vehicles(vin), add foreign key (cust_id) references customers(cust_id);
```

```
#Job_Services Table - Change datatype and add key constraints
alter table job_services modify column job_id varchar(255), modify column service_id varchar(255);
alter table job_services add primary key(job_id, service_id);
alter table job_services add foreign key(job_id) references jobs(job_id), add foreign key(service_id) references services(service_id);

#Invoice Table - Change datatype and add key constraints
alter table invoice modify column inv_no varchar(255), modify column job_id varchar(255), modify column inv_amt_total double;
alter table invoice add primary key(inv_no);
alter table invoice add foreign key(job_id) references jobs(job_id);
update invoice set inv_date = STR_TO_DATE(inv_date, "%m/%d/%Y %H:%i:%s");

#Employee_Job Table - Change datatype and add key constraints
delete from employee_job where emp_id = "emp_id";
alter table employee_job modify column job_id varchar(255), modify column emp_id varchar(255);
alter table employee_job add primary key(job_id, emp_id);
alter table employee_job add foreign key(job_id) references jobs(job_id), add foreign key(emp_id) references employee(emp_id);

#Job_Consumables Table - Change datatype and add key constraints
alter table job_consumables modify column job_id varchar(255), modify column consum_id varchar(255);
alter table job_consumables add primary key(job_id, consum_id);
alter table job_consumables add foreign key(job_id) references jobs(job_id), add foreign key(consum_id) references consumables(consum_id);

set foreign_key_checks = 1;
```

Query1

```
#Queries:
#1. Most popular service offered
select category, services.name, count(services.service_id) as popular_service from job_services
inner join services on job_services.service_id = services.service_id
group by category, services.name order by popular_service desc;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

category	name	popular_service
Tires	New Tires	162
Tires	Aluminum Tig Welding	155
Collision Repair	Paint (custom)	148
Exhaust	Dual Mufflers	146
Auto Repair	Diagnostics	143
Exhaust	Bolt on Kits	143
Suspension	2WD Lift Kits	142
Collision Repair	Auto Parts Replacement	142
Auto Repair	AC Repair	142

Query2

```
103
104     #2. Most popular service type offered
105 •   select category, count(services.service_id) as popular_service from job_services
106     inner join services on job_services.service_id = services.service_id
107     group by category order by popular_service desc;
108
109
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

category	popular_service
Suspension	782
Auto Repair	690
Tires	681
Exhaust	535
Collision Repair	529
Wheels	396

Query3

```
61
62
63 •   select c.fname, c.lname, count(c.cust_id) as frequent_customer from jobs as j
64     inner join vehicle as v on j.vin = v.vin inner join customers as c on c.cust_id = j.cust_id
65     group by c.cust_id order by frequent_customer desc limit 5;
66
67
68
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

fname	lname	frequent_customer
Charlie	Miller	4
Edgar	Henderson	3
Honey	Davis	3
Savana	Scott	3
Jessica	Perry	3

Query4

```
121
122     #4. Number of jobs that are open longer than average times for a job
123 •   select count(job_id) from jobs where status = "Open" and
124     datediff((select max(end_datetime) from jobs), start_datetime) > (select avg(datediff(jobs.end_datetime, jobs.start_datetime)) from jobs);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

count(job_id)
145

Query5

```
126    #5. Cost per hour for every service
127    select service_id, category, name, round(sum(base_rate)/sum(service_hours),2) as cost_hr
128    from
129    (select s.service_id, service_hours, category, name, base_rate
130    from
131    job_services as js
132    inner join
133    services as s
134    on s.service_id = js.service_id) as a
135    group by service_id, category, name
136    order by cost_hr desc
137    limit 10;
138
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

service_id	category	name	cost_hr
S1_004	Collision Repair	Auto Parts Replacement	668.16
S1_003	Collision Repair	Autobody	528.66
S4_006	Suspension	2WD Lift Kits	436.69
S3_002	Auto Repair	Diagnostics	318.87
S4_004	Suspension	Alignment	298.09
S4_005	Suspension	4WD Lift Kits	275.95
S6_001	Wheels	New Rims	255.84
S6_003	Wheels	Luxury Rims	228.3
S2_002	Exhaust	Bolt on Kits	217.27
S5_004	Tires	Rim Repair	172.9

Query6

```
67   SELECT make, COUNT(make) As Numvehicles, /*Count frequency of cars with each make*/
68   count(make)*100/(SELECT count(*) FROM jobs) as percent_vehicles,
69   sum(inv_amt_total)*100/(SELECT sum(inv_amt_total) FROM invoice) as percent_invoices,
70   sum(inv_amt_total) as sum_invoices,
71   avg(inv_amt_total) as avg_invoices,
72   max(inv_amt_total) as max_invoices
73   FROM (select a.job_id, a.vin, a.end_datetime, a.start_datetime, a.times_reopened,
74   b.inv_discount, b.inv_amt_total, b.service_miles, b.inv_tax_total, c.make, c.model, c.year
75   from jobs a left join invoice b on a.job_id=b.job_id left join vehicle c on a.vin=c.vin) as ab
76   GROUP BY make
77   order by avg_invoices desc;
78
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

make	Numvehicles	percent_vehicles	percent_invoices	sum_invoices	avg_invoices	max_invoices
Honda	73	4.8667	5.7999	76660	1179.3846	4950
Jeep	97	6.4667	7.6206	100725	1106.8681	3490
Nissan	112	7.4667	7.6679	101350	1078.1915	4575
BMW	104	6.9333	7.3202	96755	1029.3085	5110
Dodge	112	7.4667	8.0741	106720	997.3832	4750
Saab	108	7.2000	7.1920	95060	990.2083	4775
Ford	128	8.5333	8.6578	114435	986.5086	6155
Volkswagen	99	6.6000	6.6022	87265	980.5056	3680
Toyota	106	7.0667	6.9801	92260	971.1579	3470
Accura	96	6.4000	6.1343	81080	931.9540	4000
Subaru	102	6.8000	6.4517	85275	926.9022	4755
Mercedes	132	8.8000	7.9531	105120	890.8475	3930
Chevrolet	111	7.4000	6.5841	87025	879.0404	3780
Audi	120	8.0000	6.9620	92020	860.0000	4610

Query7

```
#7. Sales by Car Type - 7: Majority share (54.8667%)
select make, count(model)*100/(select count(model) as total from jobs as j
inner join vehicle as v on v.vin = j.vin) as make_share
from jobs as j inner join vehicle as v on v.vin = j.vin group by make
order by make_share desc limit 7;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

make	make_share
Mercedes	8.8000
Ford	8.5333
Audi	8.0000
Nissan	7.4667
Dodge	7.4667
Chevrolet	7.4000
Saab	7.2000

Query8

```
#8. More individuals have newer cars
select year, count(model)*100/(select count(model) as total from jobs as j inner join vehicle as v
on v.vin = j.vin) as year_share from jobs as j inner join
vehicle as v on v.vin = j.vin group by year order by year_share desc;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

year	year_share
2009	7.3333
2013	7.3333
2014	7.2000
2011	7.0667
2012	6.8000
2010	6.6667
2015	6.5333
2008	6.2667

Query9

```
#9. Car Makes that have jobs frequently reopened
select a.make, b.make_reopen*100/a.make_share as perc_reopen from (select make, count(model) as make_share
from jobs as j inner join vehicle as v on v.vin = j.vin group by make) as a
inner join (select make, count(model) as make_reopen from (select * from jobs where times_reopened >0) as j
inner join vehicle as v on v.vin = j.vin group by make) as b on a.make = b.make
order by perc_reopen desc
limit 5;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

make	perc_reopen
Honda	45.2055
Saab	37.9630
Jeep	36.0825
Subaru	35.2941
Volkswagen	32.3232

Query10

```
170
171     #10. Car Years with the highest % reopen instance.
172 •   select a.year,  b.year_reopen*100/a.year_share as perc_reopen from (select year, count(model) as year_share
173     from jobs as j inner join vehicle as v on v.vin = j.vin group by year) as a inner join
174     (select year, count(model) as year_reopen from
175     (select * from jobs where times_reopened >0) as j inner join vehicle as v on v.vin = j.vin group by year) as b on a.year = b.year
176     order by perc_reopen desc;
177
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

make	perc_reopen
Honda	45.2055
Saab	37.9630
Jeep	36.0825
Subaru	35.2941
Volkswagen	32.3232

Query11

```
207     #11. Hours spent by Contractual and Full time employees
208 •   select type, sum(hrs_emp)/(select datediff(max(end_datetime) , min(start_datetime)) from jobs) as thours from
209     (select js.job_id, emp_id, type, service_hours/emp_count as hrs_emp from
210     (select job_id, ejc.emp_id, emp_count, type from
211     (select ej.job_id, emp_id, emp_count from
212     (select * from employee_job) as ej inner join
213     (select job_id, count(emp_id) as emp_count from employee_job group by job_id) as ec on ej.job_id = ec.job_id) as ejc
214     inner join
215     (select emp_id, type from employee) as e on e.emp_id = ejc.emp_id) as ejct
216     inner join
217     (select job_id, sum(service_hours) as service_hours from job_services group by job_id) as js on ejct.job_id = js.job_id) as eh
218     group by type;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

type	thours
Contract	91.1923904865002
Permanent	58.302306483201235

Stage 3: Hypothesizing Business Recommendations and Insights

Human resource planning and optimization

- Employees must be professionally trained to handle jobs related to Mercedes, Ford, and Audi cars, as they account for ~25 per cent of the jobs. Additionally, Mr. Lopez can consider exclusive customer service partnerships with these makes to increase credibility of his shop and attract cautious customers
- Employees need to be educated about regulatory challenges for cars made in the years 2010 and 2015, as these cars account for ~35% of the vehicles coming in for service
- The shop must consider hiring more full-time employees as contractual employees (7) are working ~56% more hours (~91 hours per day) as compared to Full-Time employees (11) (~58 hours per day), and since they are paid hourly, Mr. Lopez can highly minimize the salary he pays, thus creating an opportunity to increase profit
- Even though Honda cars have the least number of cars coming in for service, they have the highest re-opening rate – Employees must be trained specifically to handle cases related to Honda cars

Inventory Management and Order Planning

- Frequent parts such as exhausts, ball pins, and engine belts for Ford, Mercedes, and Audi must be ordered regularly to minimize service time

Stage 3: Hypothesizing Business Recommendations and Insights

Building Customer Relationships

- Ford, BMW, and Dodge vehicles account for the maximum invoice amounts, thus the auto shop can consider a way to strengthen relationship with these customers such as offering discounts, to ensure they keep coming back