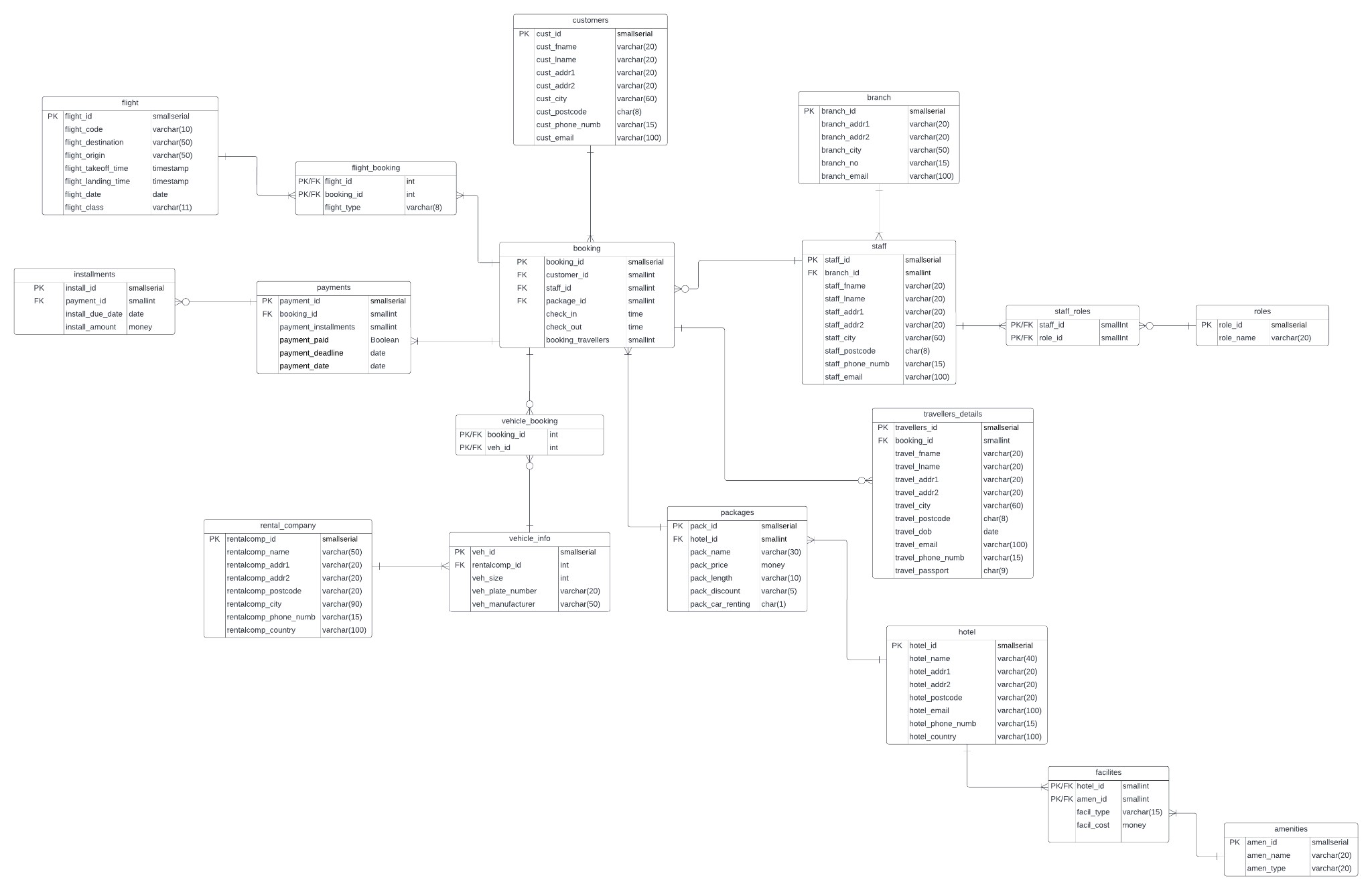
# EERD (Enhanced Entity Relationship Diagram)



# Assumptions

* A staff member could have multiple roles/responsibilities eg. Manager is also an accountant.
* Multiple packages can have the same hotel and flight.
* If a customer doesn't pay within a certain timeframe booking is cancelled.
* If vehicle booking is not available in a package then there will be no entry in the vehicle\_booking table for that booking.
* Multiple payments can be made for a single booking.
* Some roles (e.g cleaners) can’t book customers in or out hence why there is a zero or many relationship between staff and booking. - There is a zero or many relationship between booking and travellers details because some customers may come alone.
* There is a zero or many relationship between payments and installments because some payments may not have any installment (a customer can pay everything on the spot).

# Data Dictionary

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | booking |  |  |
| ***Attribute\_Name*** | ***KEY*** | ***INDEX*** | ***Data Type & Size*** | ***Domains***  ***& Constraints*** | ***FK Reference*** | ***Description*** |
| booking\_id | PK |  | smallserial | Not Null, unique |  |  |
| customer\_id | FK | Y | smallint | Not Null | customer.cust\_id |  |
| staff\_id | FK | Y | smallint | Not Null | staff.staff\_id |  |
| package\_id | FK | Y | smallint | Not Null | package.pack\_id |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| check\_in |  | |  | | time | Not Null | |  | | Exact check in time |
| check\_out |  | |  | | time | Not Null | |  | | Exact check out time |
| booking\_travellers |  | |  | | smallint | Not Null | |  | | Number of travellers booked in by customer |
|  |  |  | |  | | customers |  | |  | |
| ***Attribute\_Name*** | ***KEY*** | ***INDEX*** | | ***Data Type & Size*** | | ***Domains***  ***& Constraints*** | ***FK Reference*** | | ***Description*** | |
| cust\_id | PK |  | | smallserial | | Not Null, unique |  | |  | |
| cust\_fname |  |  | | varchar(20) | | Not Null |  | |  | |
| cust\_lname |  |  | | varchar(20) | | Not Null |  | |  | |
| cust\_addr1 |  |  | | varchar(20) | | Not Null |  | |  | |
| cust\_addr2 |  |  | | varchar(20) | |  |  | |  | |
| cust\_city |  |  | | varchar(60) | | Not Null |  | |  | |
| cust\_postcode |  |  | | char(8) | | Not Null |  | |  | |
| cust\_phone\_numb |  |  | | varchar(15) | | Not Null |  | |  | |
| cust\_email |  |  | | varchar(100) | | Not Null |  | |  | |
|  |  |  | |  | | packages |  | |  | | |
| ***Attribute\_Name*** | ***KEY*** | ***INDEX*** | | ***Data Type & Size*** | | ***Domains***  ***& Constraints*** | ***FK Reference*** | | ***Description*** | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| pack\_id | PK |  | smallserial | Not Null, unique |  | |  | |
| hotel\_id | FK |  | smallint | Not Null | hotel.hotel\_id | |  | |
| pack\_name |  |  | varchar(30) | Not Null |  | |  | |
| pack\_price |  |  | money | Not Null |  | |  | |
| pack\_length |  |  | varchar(10) |  |  | |  | |
| pack\_discount |  |  | varchar(5) | Not Null |  | |  | |
| pack\_car\_renting |  |  | char(1) | Not Null |  | | It reflects if a vehicle is assigned to a package | |
|  |  |  |  | hotel | |  | |  |
| ***Attribute\_Name*** | ***KEY*** | ***INDEX*** | ***Data Type & Size*** | ***Domains***  ***& Constraints*** | | ***FK Reference*** | | ***Description*** |
| hotel\_id | PK |  | smallserial | Not Null, unique | |  | |  |
| hotel\_name |  | Y | varchar(40) | Not Null | |  | |  |
| hotel\_addr1 |  |  | varchar(20) | Not Null | |  | |  |
| hotel\_addr2 |  |  | varchar(20) |  | |  | |  |
| hotel\_postcode |  |  | varchar(20) | Not Null | |  | |  |
| hotel\_email |  |  | varchar(100) | Not Null | |  | |  |
| hotel\_phone\_numb |  |  | varchar(15) | Not Null | |  | |  |
| hotel\_country |  |  | varchar(100) | Not Null | |  | |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | facilities |  | | |  | | | |
| ***Attribute\_Name*** | ***KEY*** | ***INDEX*** | ***Data Type & Size*** | ***Domains***  ***& Constraints*** | ***FK Reference*** | | | ***Description*** | | | |
| hotel\_id | FK |  | smallint | Not Null | hotel.hotel\_id | | |  | | | |
| amen\_id | FK |  | smallint | Not Null | amenities.amen\_id | | |  | | | |
| facil\_type |  |  | varchar(15) | Not Null |  | | |  | | | |
| facil\_cost |  |  | money | Not Null |  | | | Any extra cost associated with the facility | | | |
|  |  |  |  | amenities | |  | | |  | |
| ***Attribute\_Name*** | ***KEY*** | ***INDEX*** | ***Data Type & Size*** | ***Domains***  ***& Constraints*** | | ***FK Reference*** | | | ***Description*** | |
| amen\_id | PK |  | smallserial | Not Null, unique | |  | | |  | |
| amen\_name |  |  | varchar(20) | Not Null | |  | | |  | |
| amen\_type |  |  | varchar(20) | Not Null | |  | | | Clothes, Product, etc | |
|  |  |  |  | payments | | |  | | |  |
| ***Attribute\_Name*** | ***KEY*** | ***INDEX*** | ***Data Type & Size*** | ***Domains***  ***& Constraints*** | | | ***FK Reference*** | | | ***Description*** |
| payment\_id | PK |  | smallserial | Not Null, unique | | |  | | |  |
| booking\_id | FK |  | smallint | Not Null | | | booking.booking\_id | | |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| payment\_installment s |  |  | smallint | Not Null |  | | | |  | | | |
| payment\_paid |  |  | boolean | Not Null |  | | | |  | | | |
| payment\_deadline |  |  | date | Not Null |  | | | |  | | | |
| payment\_date |  |  | date |  |  | | | | The date when the payment was made | | | |
|  |  |  |  | installments | |  | | | |  | | |
| ***Attribute\_Name*** | ***KEY*** | ***INDEX*** | ***Data Type & Size*** | ***Domains***  ***& Constraints*** | | ***FK Reference*** | | | | ***Description*** | | |
| install\_id | PK |  | smallserial | Not Null, unique | |  | | | |  | | |
| payment\_id | FK | Y | smallint | Not Null | | payments.payment\_id | | | |  | | |
| install\_due\_date |  |  | date | Not Null | |  | | | |  | | |
| install\_amount |  |  | money | Not Null | |  | | | |  | | |
|  |  |  |  | staff | | |  | | | |  | |
| ***Attribute\_Name*** | ***KEY*** | ***INDEX*** | ***Data Type & Size*** | ***Domains***  ***& Constraints*** | | | ***FK Reference*** | | | | ***Description*** | |
| staff\_id | PK |  | smallserial | Unique, Not Null | | |  | | | |  | |
| branch\_id | FK |  | smallint | Not Null | | | branch.branch\_id | | | |  | |
| staff\_fname |  |  | varchar(20) | Not Null | | |  | | | |  | |
| staff\_lname |  |  | varchar(20) | Not Null | | |  | | | |  | |
| staff\_addr1 |  |  | varchar(20) | Not Null | | |  | | | |  | |
| staff\_addr2 |  |  | varchar(20) |  | | |  | | | |  | |
| staff\_city |  |  | varchar(60) | Not Null | | |  | | | |  | |
| staff\_postcode |  |  | char(8) | Not Null | | |  | | | |  | |
| staff\_phone\_numb | AK |  | varchar(15) | Not Null, Unique | | |  | | | |  | |
| staff\_email |  |  | varchar(100) | Not Null | | |  | | | |  | |
|  |  |  |  | branch | | | |  | | | |  |
| ***Attribute\_Name*** | ***KEY*** | ***INDEX*** | ***Data Type & Size*** | ***Domains***  ***& Constraints*** | | | | ***FK Reference*** | | | | ***Description*** |
| branch\_id | PK |  | smallserial | Unique, Not Null | | | |  | | | |  |
| branch\_addr1 |  |  | varchar(20) | Not Null | | | |  | | | |  |
| branch\_addr2 |  |  | varchar(20) | Not Null | | | |  | | | |  |
| branch\_city |  |  | varchar(50) | Not Null | | | |  | | | |  |
| branch\_no |  |  | varchar(15) | Not Null | | | |  | | | |  |
| branch\_email |  |  | varchar(100) | Not Null | | | |  | | | |  |

roles

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Attribute\_Name*** | ***KEY*** | | | ***INDEX*** | | ***Data Type & Size*** | | | ***Domains***  ***& Constraints*** | | | ***FK Reference*** | | | ***Description*** | | | | | |
| role\_id | PK | | |  | | smallserial | | | Unique, Not Null | | |  | | |  | | | | | |
| role\_name |  | | |  | | varchar(20) | | | Not Null | | |  | | |  | | | | | |
|  |  | | |  | |  | | | staff\_roles | | | |  | | | |  | | |
| ***Attribute\_Name*** | ***KEY*** | | | ***INDEX*** | | ***Data Type & Size*** | | | ***Domains***  ***& Constraints*** | | | | ***FK Reference*** | | | | ***Description*** | | |
| staff\_id | FK | | |  | | smallint | | | Unique, Not Null | | | | staff.staff\_id | | | |  | | |
| role\_id | FK | | |  | | smallint | | | Not Null | | | | roles.role\_id | | | |  | | |
|  | |  | |  | |  | | | rental\_company | | |  | | | | |  | |
| ***Attribute\_Name*** | | ***KEY*** | | ***INDEX*** | | ***Data Type & Size*** | | | ***Domains***  ***& Constraints*** | | | ***FK Reference*** | | | | | ***Description*** | |
| rentalcomp\_id | | PK | |  | | smallserial | | | Unique, Not Null | | |  | | | | |  | |
| rentalcomp\_name | |  | |  | | varchar(50) | | | Not Null | | |  | | | | |  | |
| rentalcomp\_addr1 | |  | |  | | varchar(20) | | | Not Null | | |  | | | | |  | |
| rentalcomp\_addr2 | |  | |  | | varchar(20) | | |  | | |  | | | | |  | |
| rentalcomp\_postcode | |  | |  | | varchar(20) | | |  | | |  | | | | |  | |
| rentalcomp\_city | |  | |  | | varchar(90) | | | Not Null | | |  | | | | |  | |
| rentalcomp\_phone\_n | |  | |  | | varchar(15) | | | Not Null | | |  | | | | |  | |
| umb | |  | |  | |  | | |  | | |  | | | | |  | |
| rentalcomp\_country | |  | |  | | varchar(100) | | | Not Null | | |  | | | | |  | |
|  | |  | |  | |  | | | vehicle\_info | |  | | | | |  | | | |
| ***Attribute\_Name*** | | ***KEY*** | | ***INDEX*** | | ***Data Type & Size*** | | | ***Domains***  ***& Constraints*** | | ***FK Reference*** | | | | | ***Description*** | | | |
| veh\_id | | PK | |  | | smallserial | | | Unique, Not Null | |  | | | | |  | | | |
| rentalcomp\_id | | FK | |  | | int | | | Not Null | | Rental\_company.rentalcomp\_id | | | | |  | | | |
| veh\_size | |  | |  | | int | | | Not Null | |  | | | | | How many seats a car has | | | |
| veh\_plate\_number | |  | |  | | varchar(20) | | | Not Null | |  | | | | |  | | | |
| veh\_manufacturer | |  | |  | | varchar(50) | | | Not Null | |  | | | | |  | | | |
|  | | |  | |  | |  | vehicle\_booking | | | | | |  | | | |  | |
| ***Attribute\_Name*** | | |  | | ***KEY*** | | ***INDEX*** | ***Data Type & Size*** | | ***Domains***  ***& Constraints*** | | | | ***FK Reference*** | | | | ***Description*** | |
| booking\_id | | | FK | |  | |  | int | | Unique, Not Null | | | | booking.booking\_id | | | |  | |
| veh\_id | | | FK | |  | |  | int | | Not Null | | | | vehcle\_info.veh\_id | | | |  | |

flight

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Attribute\_Name*** | | ***KEY*** | ***INDEX*** | | ***Data Type & Size*** | | ***Domains***  ***& Constraints*** | | ***FK Reference*** | | ***Description*** | |
| flight\_id | | PK |  | | smallserial | | Unique, Not Null | |  | |  | |
| flight\_code | |  |  | | varchar(10) | | Not Null | |  | |  | |
| flight\_destination | |  |  | | varchar(50) | | Not Null | |  | |  | |
| flight\_origin | |  |  | | varchar(50) | | Not Null | |  | | The airport from which the customer takes off | |
| flight\_takeoff\_time | |  |  | | timestamp | | Not Null | |  | |  | |
| flight\_landing\_time | |  |  | | timestamp | | Not Null | |  | |  | |
| flight\_date | |  |  | | date | | Not Null | |  | |  | |
| flight\_class | |  |  | | varchar(10) | | Not Null | |  | |  | |
|  |  | | |  | | flight\_booking | | | |  | |  | |
| ***Attribute\_Name*** | ***KEY*** | | | ***INDEX*** | | ***Data Type & Size*** | | ***Domains***  ***& Constraints*** | | ***FK Reference*** | | ***Description*** | |
| flight\_id | FK | | |  | | int | | Not Null | | flight.flight\_id | |  | |
| booking\_id | FK | | |  | | int | | Not Null | | booking.booking\_id | |  | |
| flight\_type |  | | |  | | varchar(8) | | Not Null | |  | | Outbound/Return | |

travellers\_details

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***Attribute\_Name*** | ***KEY*** |  | ***INDEX*** | ***Data Type & Size*** | ***Domains***  ***& Constraints*** | ***FK Reference*** | ***Description*** |
| travellers\_id | PK |  |  | smallserial | Unique, Not Null |  |  |
| booking\_id | FK |  |  | smallint | Not Null | booking.booking\_id |  |
| travel\_fname |  |  |  | varchar(20) | Not Null |  |  |
| travel\_lname |  |  |  | varchar(20) | Not Null |  |  |
| travel\_addr1 |  |  |  | varchar(20) | Not Null |  |  |
| travel\_addr2 |  |  |  | varchar(20) |  |  |  |
| travel\_city |  |  |  | varchar(60) | Not Null |  |  |
| travel\_postcode |  |  |  | char(8) | Not Null |  |  |
| travel\_dob |  |  |  | date | Not Null |  |  |
| travel\_email |  |  |  | varchar(100) | Not Null |  |  |
| travel\_phone\_numb |  |  |  | varchar(15) | Not Null |  |  |
| travel\_passport |  | Y |  | char(9) | Not Null |  |  |

# Optimization

Based on the T.A and discussion; booking, customer and payments table would need to be properly optimised and we plan on going about this with these techniques.

The first technique we used were indexes. Indexes when used rightly can greatly reduce running time of queries by reducing the amount of data that needs to be accessed to answer the queries (Chaudhuri & Narasayya 2004).

Before applying an index to a table, the table would most likely not be discernible(Chartio, nd). This means when a query is run every single row would have to be searched linearly(Chartio, nd) meaning if you have 10 million rows the query would go over the data 10 million times. What an index is is simply a pointer to data in a table, An index in a database is very similar to an index in the back of a book (Tutorialspoint, nd).

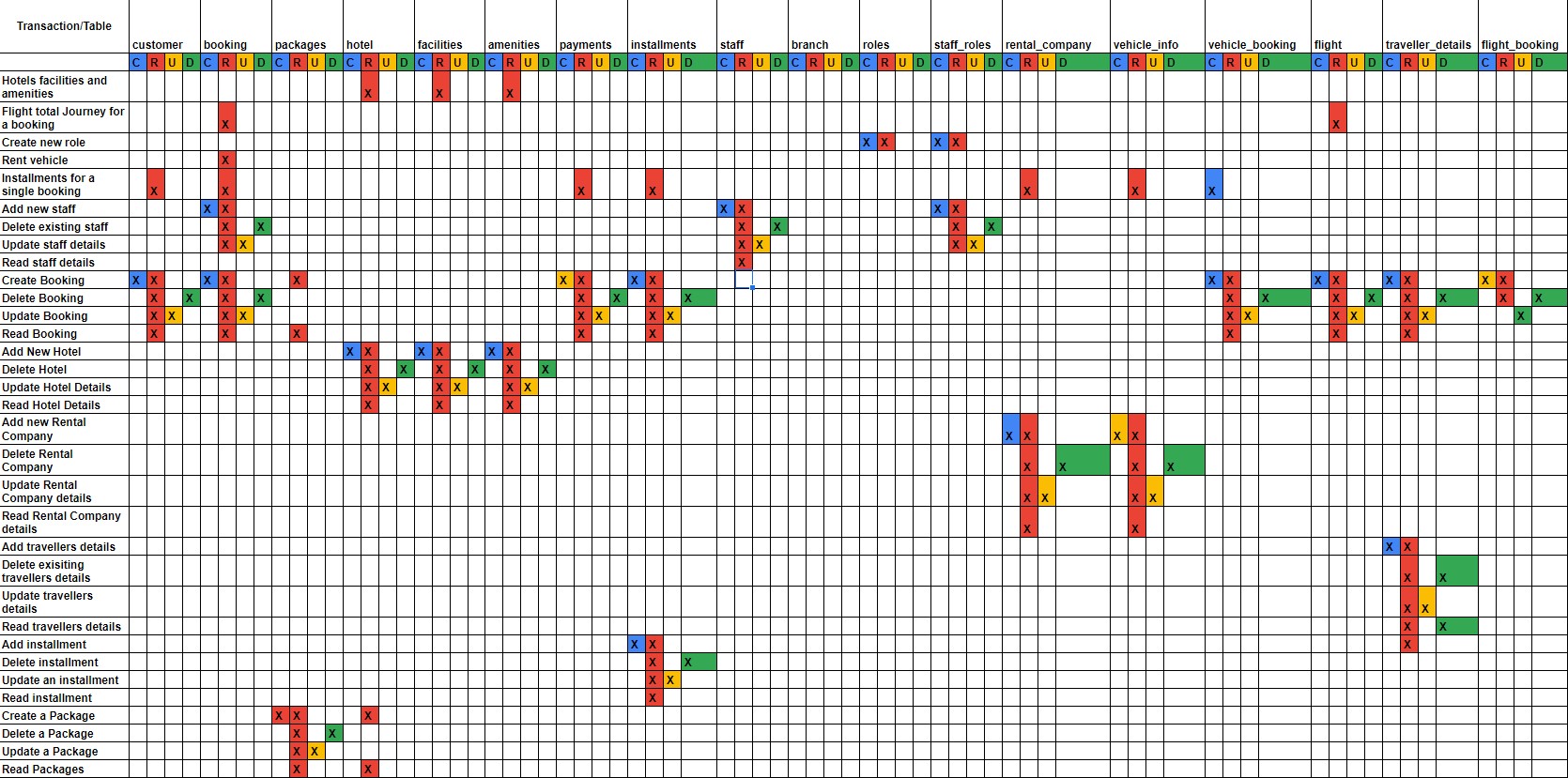
Indexes consumes space and constantly needs to be kept up to date(Chaudhuri & Narasayya, 2004). However, we don’t intend on overpopulating our database with indexes. To combat this workload we would strategically place indexes where we know it would be needed. For example the booking table, this table would hold information from bookings made. Which implies that every single day this table would be called upon for a task. Hence, creating indexes on this table would ensure maximum capacity any time it’s used.

We considered the use of correct data types. Data types are what determines the type of data that can be stored in a database table column(Packt, nd). We made sure we selected data types for each column appropriate for the data stored in that column. We compared fixed- and varying-length character data types then selected which would be more efficient per situation. An example is the cust\_postcode on the customer table which because we know the maximum length of postcodes needed we set it to Char(8) in order to preserve memory. Another example is the customer id for the customer table created which we assigned a serial data type which can hold a maximum of 2,147,483,647(IBM, 2022) values. Over a couple of years considering that this company is international, it is almost certain that this type would be sufficient without a need to constantly update.

We also implemented specific data types to ensure that specific data pertaining to the data type is inputted. For example booking holds a column check\_in, that holds a time and date. We set this as a Timestamp to ensure that we get a time and date and not a name. This would ensure consistency across the whole database and would make querying data easier(Packt, nd).

Columns were carefully picked, we decided what should be NULL or NOT NULL. Anywhere we used NULL it was certain that that was the most efficient way of going about it. Setting columns as NOT NULL makes operations faster, by enabling better use of indexes and eliminating overhead for testing whether each value is NULL. You also save some storage space, one bit per column(MySQL, nd).

# Transaction Analysis Matrix



# Business Related Queries With the Outputs

List the travellers details that will check in within a certain timeframe

SELECT

CONCAT(travel\_fname,' ' ,travel\_lname) AS "Travellers Name",

CONCAT\_WS(' | ', travel\_addr1, travel\_addr2, travel\_city, travel\_postcode) AS "Travellers Address",

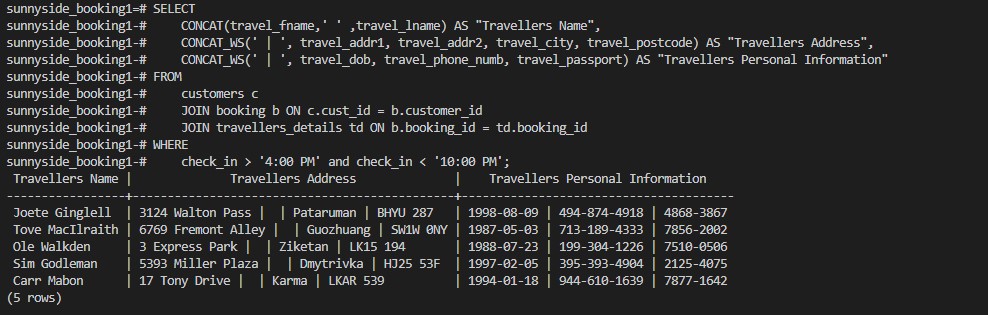
CONCAT\_WS(' | ', travel\_dob, travel\_phone\_numb, travel\_passport) AS "Travellers Personal Information"

FROM customers c

JOIN booking b ON c.cust\_id = b.customer\_id

JOIN travellers\_details td ON b.booking\_id = td.booking\_id

WHERE check\_in > '4:00 PM' and check\_in < '10:00 PM';



Query that count how many bookings are attached to a single flight

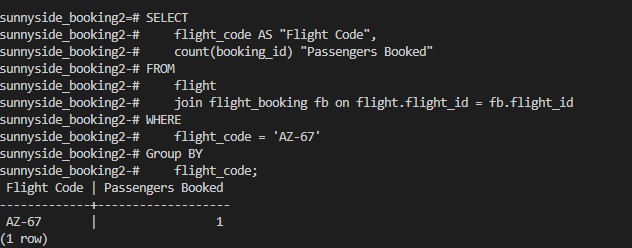
SELECT flight\_code AS "Flight Code", count(booking\_id) "Passengers Booked"

FROM flight

join flight\_booking fb on flight.flight\_id = fb.flight\_id

Where

Flight\_code = ‘AZ-67’;



Query that prints out all installments that are part of a single booking

SELECT

b.booking\_id AS "Booking ID", concat(cust\_fname, ' ', cust\_lname) AS "Booker Full Name", concat (cust\_phone\_numb, ' | ', cust\_email) AS "Booker Contact Details", install\_id AS "Installment ID", install\_amount AS "Installment Amount", install\_due\_date AS "Due dates"

FROM customers c

JOIN booking b ON c.cust\_id = b.customer\_id

JOIN payments p ON b.booking\_id = p.booking\_id

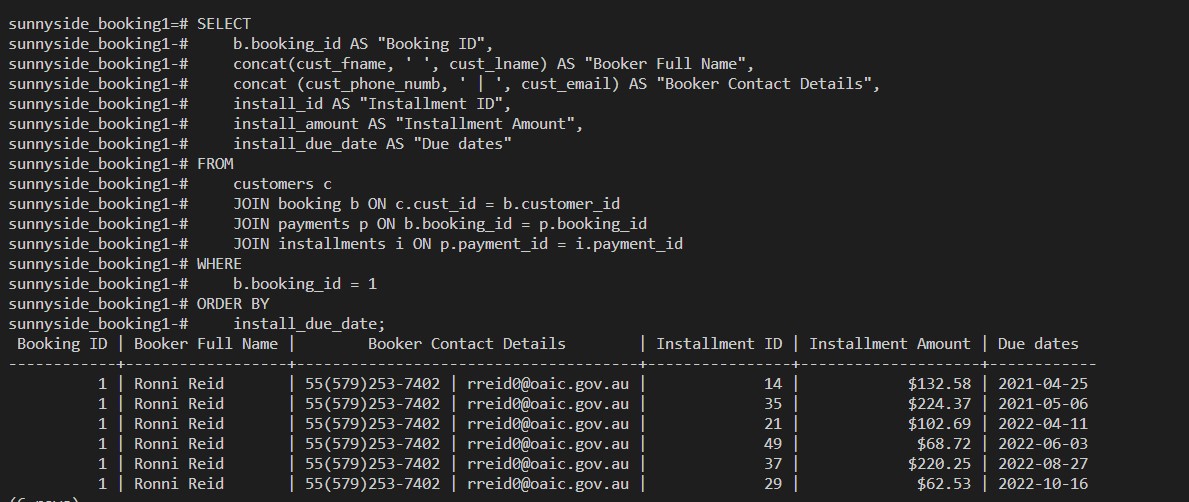
JOIN installments i ON p.payment\_id = i.payment\_id

WHERE

b.booking\_id = 1

ORDER BY

Install\_due\_date;



Query that calculates the total journey of a flight and returns the details

SELECT concat(cust\_fname, ' ', cust\_lname) AS "Booker Full Name",

concat(cust\_phone\_numb, ' | ', cust\_email) AS "Booker Contact Details",

flight\_code AS "Flight Code", flight\_date AS "Flight Date", flight\_type AS "Flight Type", flight\_class AS "Flight Class",

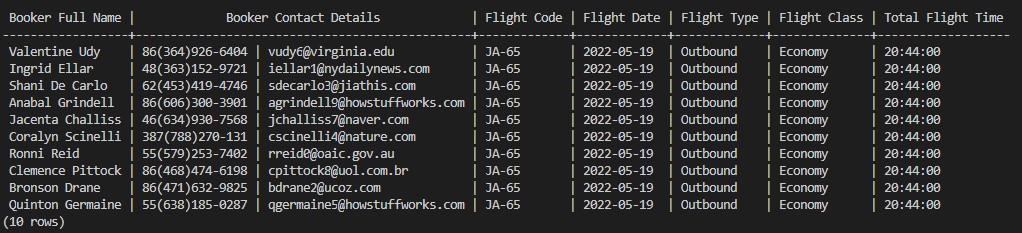
SUM(flight\_landing\_time - flight\_takeoff\_time) AS "Total Flight Time" from flight f

join flight\_booking fb on f.flight\_id = fb.flight\_id

JOIN booking b ON fb.booking\_id = b.booking\_id where

b.customer\_id = 1

GROUP BY flight\_code, flight\_date, flight\_type, Flight\_class, cust\_fname, cust\_lname, cust\_phone\_numb, cust\_email;



A query that returns the installment deadline days for customers (On a monthly basis) in order to send a light remainder to them to pay.

SELECT concat(cust\_fname, ' ', cust\_lname) AS "Booker Full Name", concat (cust\_phone\_numb, ' | ', cust\_email) AS "Booker Contact Details", p.booking\_id AS "Booking ID", install\_due\_date AS "Instalment Due Date", install\_amount AS "Amount to be Paid" from customers c join booking b on c.cust\_id = b.customer\_id join payments p on b.booking\_id = p.booking\_id join installments i on p.payment\_id = i.payment\_id

WHERE p.payment\_id in (

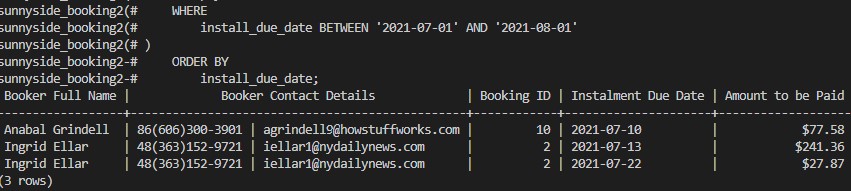
SELECT booking\_id

FROM payments

WHERE install\_due\_date BETWEEN '2021-07-01' AND '2021-08-01'

)

ORDER BY install\_due\_date;



# References

Chaudhuri, S., Datar, M., & Narasayya, V. (2004). Index selection for databases: A hardness study and a principled heuristic solution. IEEE transactions on knowledge and data engineering, 16(11), 1313-1323.

Cote, C. (2021). *5 Principles of Data Ethics for Business*. [online] Business Insights - Blog. Available at: <https://online.hbs.edu/blog/post/data-ethics>.

Fixed- and Varying-Length Character Data Types. (2022, September 28). www.ibm.com. <https://www.ibm.com/docs/en/informix-servers/14.10?topic=types-fixed-varying-length-character-data>

How Does Indexing Work. (n.d.). Chartio. <https://chartio.com/learn/databases/how-does-indexing-work/>

MySQL :: MySQL 8.0 Reference Manual :: 8.4.1 Optimizing Data Size. (n.d.). Dev.mysql.com.

<https://dev.mysql.com/doc/refman/8.0/en/data-size.html>

Packtpub.com. (2022).

[https://subscription.packtpub.com/book/networking-and-servers/9781782172550/1/ch01lvl1sec14/the-importance-of-choosing-the-appropriate-d ata-type#:~:text=A%20data%20type%20determines%20the](https://subscription.packtpub.com/book/networking-and-servers/9781782172550/1/ch01lvl1sec14/the-importance-of-choosing-the-appropriate-data-type#:~:text=A%20data%20type%20determines%20the)

Schlackl, F., Link, N., & Hoehle, H. (2022). Antecedents and Consequences of Data Breaches: A Systematic Review. Information & Management, 103638.

Support, A. (2020). *8 Tips For Ensuring Your Employees Are Handling Data Securely | Amazing Support*. [online] www.amazingsupport.co.uk.

Available at: https://www.amazingsupport.co.uk/8-tips-for-ensuring-your-employees-are-handling-data-securely/ [Accessed 9 Dec. 2022].

SQL - Indexes - Tutorialspoint. (n.d.). [www.tutorialspoint.com](http://www.tutorialspoint.com/).

<https://www.tutorialspoint.com/sql/sql-indexes.htm>

UK Government (2018). Data Protection Act 2018. [online] Legislation.gov.uk. Available at: <https://www.legislation.gov.uk/ukpga/2018/12/contents/enacted>.