Background

This database system stores data for a tourism company called "Sky High Travel Co.". The aim of this database is to assign and organize tours, tour guides and customers together, and manage each of the tour trips, as well as storing important data about the tours, customers and guides in a way that prevents redundant data and maintain consistency, as well as maintaining data integrity. This database system also brought some additional benefits such as improving the speed of data access through the use of SQL, and improving the quality of information through the use of reports, thus improving the quality of decision making.
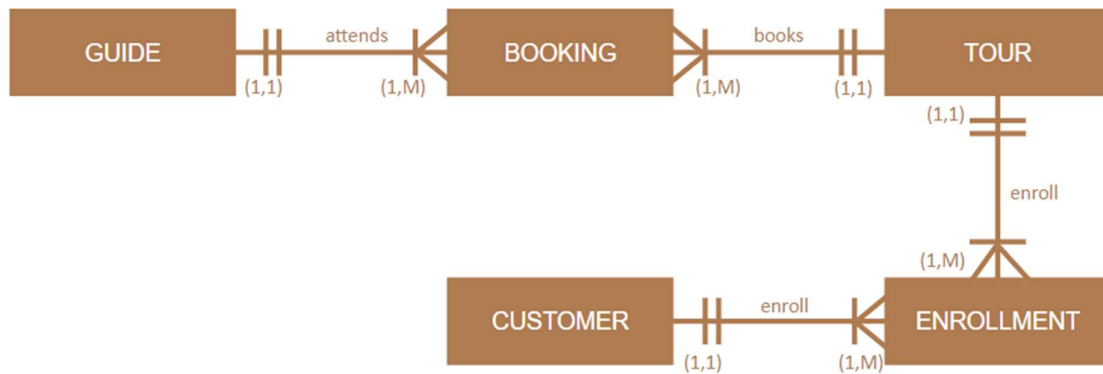

Figure 1: ER Diagram (Conceptual Model)

In this database model, there are five entities, includes CUSTOMER, GUIDE, ENROLLEMNT, BOOKING and TOUR. CUSTOMER provides all the detailed information of a customer. GUIDE provides all the detailed information of a tour guide. TOUR provides all the information of a tour plan, which can be used multiple times with different start dates. ENROLLMENT assigns TOURs and CUSTOMERs together to record a one-time trip with a start date. BOOKING assigns a tour GUIDE with a TOUR and adds additional information about the trip.

Business Rule

There are eight basic business rules in this database, relating the 5 entities together.

i.   Each tour GUIDE attends BOOKINGs.

ii.   Each BOOKING is attended by a tour GUIDE.

iii.   Each BOOKING books only one TOUR.

iv.   Each TOUR is booked by many BOOKINGs.

v.   Each TOUR is enrolled by many ENROLLMENTs.

vi.   EACH ENROLLMENT enrolls a single TOUR.

vii.   Each ENROLLMENT is enrolled by a CUSTOMER.

viii.   EACH CUSTOMER enrolls many ENROLLMENTs.

Besides, the company has set up a rule that a one-time tour trip must have at least three customers and a maximum of ten customers.
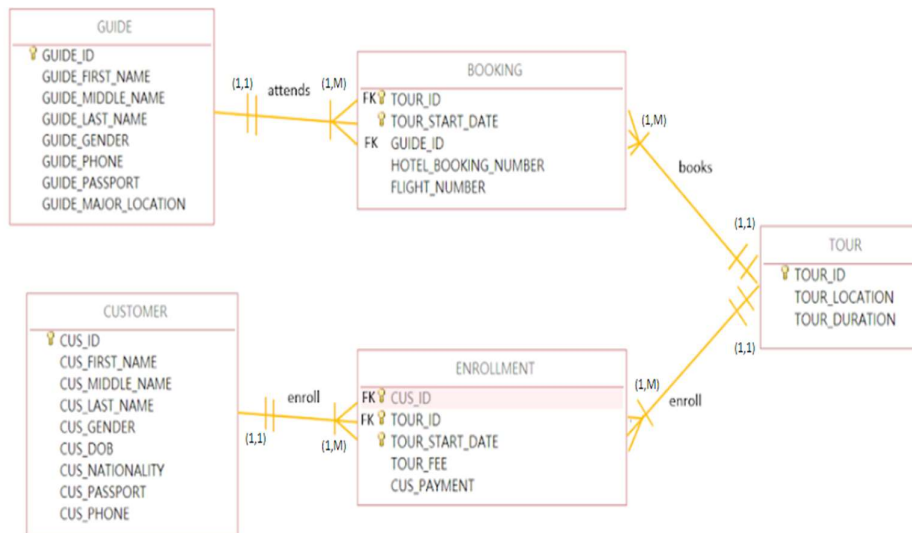

Figure 2: ER Diagram (Internal Model)

Data models:

The design of our database is normalized into the 3rd Normal Forms, which contains no repeating groups, partial dependencies, and transitive dependencies, with all the attributes being dependent on the primary key.
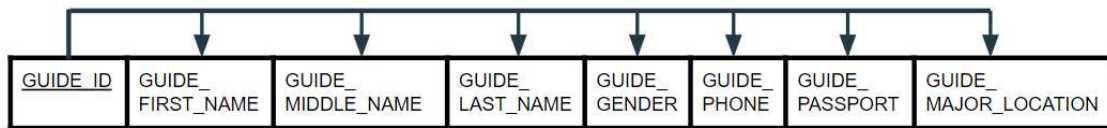
Table: GUIDE:

Figure 3: Attributes and data of GUIDE

Figure 4: 3NF of GUIDE

Referring to figure 3 and 4, the GUIDE table contained the primary key of GUIDE_ID, and contain no foreign key. GUIDE_PASSPORT, GUIDE_PHONE, and GUIDE_ID are candidate keys, but there is no secondary key for this table.

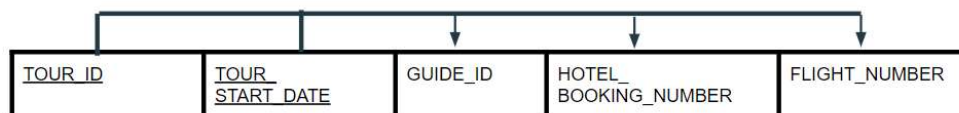***Table: BOOKING:***

Figure 5: Attributes and data of BOOKING

Figure 6: 3NF of BOOKING

Referring to figure 5 and 6, the BOOKING table contained a composed primary key of TOUR_ID and TOUR_START_DATE, and contain foreign keys of TOUR_ID and GUIDE_ID. FLIGHT_NUMBER is a possible secondary key as it can indicate the flight airline, plane type, and seat arrangement of the plane type. TOUR_START_DATE is another possible secondary key as sorting can be done to easily indicate the customers who are starting the trip soon, allowing the staff to easily contact those customers to a reminder. However, there is no candidate key for this table.

Table: TOUR:

TOUR_ID	TOUR_COUNTRY	TOUR_CITY	TOUR_DURA
1000	Australia	Sydney	15
1001	America	Las Vegas	10
1002	Japan	Okinawa	14
1003	China	Shang Hai	5
1004	Belgium	Bruges	18
1005	Austria	Vienna	14

Figure 7: Attributes and data of TOUR

Table: TOUR

TOUR_ID	TOUR_COUNTRY	TOUR_CITY	TOUR_DURATION

Figure 8: 3NF of TOUR

Referring to figure 7 and 8, the TOUR table contained a primary key of TOUR_ID, and contain no foreign key. TOUR_ID is a candidate key, but there is no secondary key for this table.

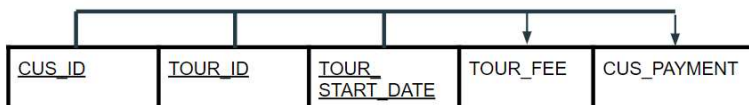*Table: ENROLLMENT:*

Table: ENROLLMENT

CUS_ID	TOUR_ID	TOUR_START_DATE	TOUR_FEE	CUS_PAYMENT

Figure 9: 3NF of ENROLLMENT

Figure 10: Attributes and data of ENROLLMENT

Referring to figure 9 and 10, the ENROLLMENT table contained a composed primary key of CUS_ID, TOUR_ID, and TOUR_START_DATE, and contain foreign keys of CUS_ID and TOUR_ID. CUS_PAYMENT is a possible secondary key as sorting can be done to indicate which customer hasn't paid the trip fee yet. TOUR_START_DATE is another possible secondary key as sorting can be done to easily indicate the customers who are starting the trip soon, allowing the staff to easily contact those customers to a reminder. However, there is no candidate key for this table.

Table: CUSTOMER:

Figure 11: Attributes and data of CUSTOMER
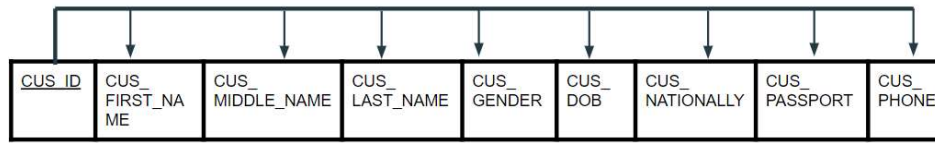
Figure 12: 3NF of CUSTOMER

Referring to figure 11 and 12, the CUSTOMER table contained a primary key of CUS_ID, and contain no foreign key. CUS_ID, CUS_PASSPORT, and CUS_PHONE are candidate keys, but there is no secondary key for this table.

Forms

Forms allow users to create a user interface in which users can enter or edit the data. Viewing, adding, and modifying data in the tables are the main usages of forms. In addition, unlike the method of direct input into the database, forms can minimize the chance of incorrect data input. Thus, minimizing the probability of data inconsistency.

Figure 13: Form A

The usage of the form "Basic information of a customer" (Figure 13) is to view and edit the basic information of a customer quickly. When giving a phone number, the relative information

of that customer (customer ID, full name, the ID of the tour joined, the start date of that tour,

the destination of that tour and the situation of payment) will be shown.

   For security concern, first, middle and last name of that customer are listed out. When

customers want to check their own information by providing phone numbers, double checking

of identity can be done easily by matching their full names. This also minimizes the chances

of mixing up people with the same first and last name.

   In addition, this form (Fig. 13) allows staff members to see the payment situation directly.

Referring to the entities CUSTOMER and ENROLLMENT is not needed.

   Besides, the basic information of the tour joined is shown. Referring to the entities TOUR

and ENROLLMENT may not be required.

Figure 14: Form B

   The form "Basic information of a tour" (Figure 14) groups the basic information of a tour,

includes the tour ID, flight number, destination, duration, tour fee, and the guide ID. This form

allows staff members to view or edit all the information related to the tour efficiently without

referring to the entities TOUR and GUIDE.

Reports

Viewing, formatting, summarizing the information in the database can be done through creating reports.

Customer in each Tour

TOUR_ID	TOUR_CITY	TOUR_START_DATE by Day	CUS_ID	CUS_FIRST_NAME	CUS_LAST_NAM	CUS_PHONE
1000	Sydney					
		Tuesday, March 12, 2019				
			10000	John	Brown	1234243252
			10011	Franklin	Field	1787945315
			10019	Mary	Porker	1234654235
		Friday, April 12, 2019				
			10007	Donald	Winters	1587878433
1001	Las Vegas					
		Thursday, April 4, 2019				
			10002	Sandra	Williams	1324786454
			10003	Jerri	Jones	1287484574
			10004	Sharon	Selph	1576846346
			10012	Alyson	Walker	1787652145
			10017	Rosalia	Jackson	1487653352
1002	Okinawa					
		Saturday, April 6, 2019				
			10001	Raymond	Carter	1524542544
			10005	Kurt	Lawson	1547546734
			10006	Phillip	Anderson	1432124746
			10013	Zachary	Shattuck	1986534756
			10014	Brenda	Genkins	1632345645
			10018	Ann	Ford	1235464657
1003	Shang Hai					
		Friday, May 31, 2019				
			10016	Jane	Williams	1546854338
1004	Bruges					
		Friday, August 2, 2019				
			10015	Christopher	Ali	1326454153
1005	Vienna					
		Wednesday, March 13, 201				
			10008	William	Caudill	1587854325
			10009	Annette	Harrell	1244676453
			10010	Jason	Webb	1541325694
			10020	James	Black	1234568765

Figure 15: Report A

The report "Customer in each Tour" (Figure 15), shows all the customers on each tour trip. This report (Fig. 15) shows which tour trip does not satisfy the cardinality of the business rules, each tour trip must have three to ten customers. For example, in TOUR_ID "10016", a trip to Shanghai on 31st May, there is only one customer. Staff can locate these trips, contacting and notifying these customers with the updated situations by phone numbers provided can be done.

Besides, this report (Figure 15) is sorted by ascending order of TOUR_START_DATE under the grouping of each TOUR. Therefore, the tour with the first TOUR_ID in the first column is

the nearest upcoming departing tour. Staff can quickly locate the upcoming trip, and notify the customers a few days before the start of their trip.

Due to different immigration policies for different countries, this report (Figure 15) is grouped by the destination of the tour. This allows staff to notify the customers for the necessary documents prepared before departing.

All the Tours for each Guide

GUIDE_ID	GUIDE_FNAME	GUIDE_LNAME	GUIDE_PHONE	TOUR_ID	TOUR_START_DATE	TOUR_COUNTRY	TOUR_CITY	TOUR_DURATION
100	Ronnie	Anderson	98765432					
				1000	12-Mar-19	Australia	Sydney	15
				1004	05-Apr-19	Belgium	Bruges	18
101	Ann	Brook	94354258					
				1005	13-Mar-19	Austria	Vienna	14
				1000	06-Apr-19	Australia	Sydney	15
				1002	30-May-19	Japan	Okinawa	14
				1004	01-Jun-19	Belgium	Bruges	18
102	Frank	Hill	12315410					
				1002	06-Apr-19	Japan	Okinawa	14
				1005	13-May-19	Austria	Vienna	14
103	Robert	Furlan	98745641					
				1003	31-May-19	China	Shang Hai	5
104	Betty	Johnson	97684654					
				1001	04-Apr-19	America	Las Vegas	10
105	John	Smith	97869753					
				1004	02-Aug-19	Belgium	Bruges	18

Figure 16: Report B

The report "All the Tours for each Guide" (Figure 16) shows all the TOURs scheduled of all tour GUIDE. As this report (Figure 16) shows the schedules of all tour guides, the management team of "Sky High Travel Co" can manage the guides easily and pay the guides according to their days of work. From the tour guides' point of view, they can know their own schedules clearly. In addition, this report (Figure 16) indicates the overlapping schedules and staff can reschedule them. For example, for GUIDE_ID 101, Ana Brook, her Japan and Belgium trips are overlapped.

Contacting and updating the guides for any emergency situations, such as flight delaying, trip canceling and fail of booking, can be done with this report (Figure 16). For example, if the tour with TOUR_ID '1000' and TOUR_START_DATE as '12-Mar-2019' is canceled, contacting the guide with GUIDE_ID '100', Ronnie Anderson' can be done.

All the TOUR_START_DATE for each TOUR_ID

TOUR ID	TOUR COUNTRY	TOUR CITY	TOUR DURATION	TOUR START DATE	GUIDE ID	HOTEL BOOKING NUMBER	FLIGHT NUMBER
1000	Australia	Sydney	15				
				12-Mar-19	100	12354	CA151
				06-Apr-19	101	27382	CX6712
1001	America	Las Vegas	10				
				04-Apr-19	104	65512	XC266
1002	Japan	Okinawa	14				
				06-Apr-19	102	45612	SD421
				30-May-19	101	84732	CT288
1003	China	Shang Hai	5				
				31-May-19	103	57842	SD455
1004	Belgium	Bruges	18				
				05-Apr-19	100	28399	KA933
				01-Jun-19	101	29921	CA991
				02-Aug-19	105	89425	NM142
1005	Austria	Vienna	14				
				13-Mar-19	101	15423	ZX233
				13-May-19	102	39992	KA827

Figure 17: Report C

The report "All the TOUR_START_DATE for each TOUR_ID" (Figure 17) shows all the scheduled trips for each tour. Customers can choose all the scheduled tours from this report (Figure 17). If there is an issue with the booking or flight of a tour, staff can indicate which guide is in charged form this report (Figure 17). Thus, allowing them to inform that guide immediately.

SQL

Three default SOL codes for different situations that may encounter in the future are designed, they are named by "Outstanding payment", "To be booked" and "Top customer".

Outstanding Payment:

```
SELECT A.CUS_ID, A.CUS_PAYMENT, A.TOUR_ID,
A.TOUR_START_DATE, A.TOUR_FEE, B.CUS_PHONE

FROM ENROLLMENT AS A, CUSTOMER AS B
WHERE A.CUS_ID=B.CUS_ID AND A.CUS_PAYMENT
= NO
ORDER BY A.TOUR_START_DATE, A.TOUR_ID,
B.CUS_ID;
```

Figure 18: SQL A

Figure 19: Result of SQL A

The first query is "Outstanding payment" (Figure 18,19). This query shows which customer has not paid yet. Also, this query is ordered by TOUR_START_DATE that staff can determine which customer needed to contact first, based on how recent the start date of the tour is.

To be booked:


```
SELECT TOUR_ID, TOUR_START_DATE, COUNT(CUS_ID) AS
NUM_OF_CUS

FROM ENROLLMENT
WHERE (TOUR_ID IN (SELECT TOUR_ID FROM BOOKING) AND
TOUR_START_DATE IN (SELECT TOUR_START_DATE FROM
BOOKING))=FALSE

GROUP BY TOUR_ID, TOUR_START_DATE
ORDER BY TOUR_START_DATE, TOUR_ID;
```

Figure 20: SQL B

Figure 21: Result of SQL B

The second query is "To be booked" (Figure 20,21). Checking which ENROLLMENT does not contain in a BOOKING can be done through this query. The start date and the number of

customers of the corresponding tour are also provided, to give extra information for booking purposes. For example, NUM_OF_CUSTOMERS give information on how many hotel rooms should be booked.

Top Customer:

```
SELECT A.CUS_ID, B.CUS_LAST_NAME,
B.CUS_PHONE, SUM (TOUR_FEE) AS TOTAL_SPENT
FROM ENROLLMENT AS A, CUSTOMER AS B

WHERE CUS_PAYMENT = YES AND A.CUS_ID =
B.CUS_ID

GROUP BY A.CUS_ID, B.CUS_LAST_NAME,
B.CUS_PHONE
ORDER BY SUM (TOUR_FEE) DESC;
```

Figure 22: SQL C

CUS_ID	CUS_LAST_	CUS_PHONE	TOTAL_SPENT
10017	Jackson	1487653352	$150,200.00
10014	Genkins	1632345645	$44,688.00
10016	Williams	1546854338	$35,420.00
10018	Ford	1235464657	$32,443.00
10011	Field	1787945315	$23,453.00
10006	Anderson	1432124746	$23,423.00
10007	Winters	1587878433	$23,420.00
10002	Williams	1324786454	$21,354.00

Figure 23: Result of SQL C

The third situation is "Top customer" (Figure 22,23). This query lists the spending of customers in descending order so that the company can know who the top-spent customer is. Information such as average spending on the tour can also be calculated based on this SQL, thus allowing the staff to modify the content of the tour, such as choice of attractions.
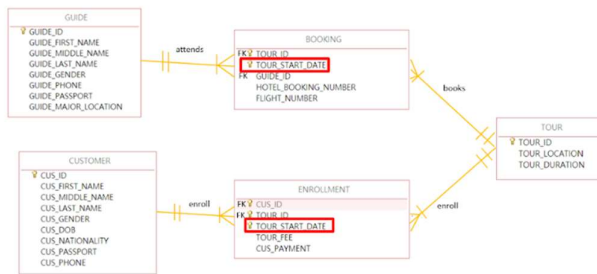
Conclusion:

Drawbacks:

Figure 24: Data Redundancy

There is a redundant attribute, TOUR_START_DATE, in this database (Figure 24). It exists in two entities at the same time without being linked together. This causes insertion, modification, and deletion anomalies. When inserting, modifying and deleting data, incorrect data input causes data inconsistency.
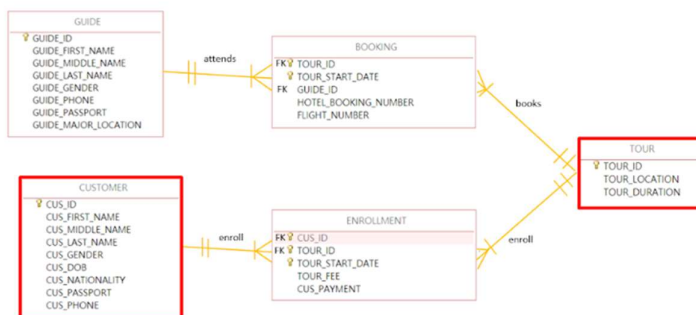

Figure 25: Cardinality

There is a cardinality relationship between each TOUR and CUSTOMERs, that each tour must have three to ten customers. However, it is impossible to create one in this model, as there is another entity of ENROLLMENT between TOUR and CUSTOMER.

Improvement:

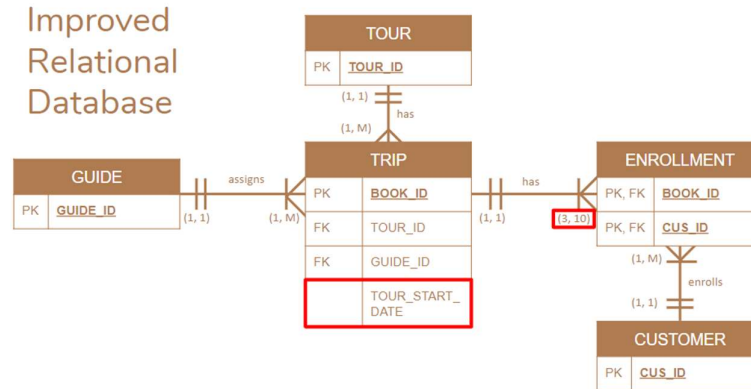There is a suggested improvement for future purposes (Figure 26).

Figure 26 Improved Relational Database

From Figure 26, for the TRIP entity, it assigns all the TOUR, GUIDE and CUSTOMER together. Since TRIP and CUSTOMER is a many-to-many relationship, another entity ENROLLMENT is created in the middle. As there is only one 'TOUR_START_DATE', the problem of data redundancy is removed. Since this model can limit the number of customers through the relationship between TRIP and ENROLLMENT, the problem of cardinality is solved.