

## Task 1:

**Branch** (branchNo, bName, bStreetNo, bStreetName, bPostCode, bState, numberEmployees)

```
> CREATE table Branch (  
branchNo CHAR(3) Primary key,  
bName TEXT,  
bStreetNo TEXT,  
bStreetName TEXT,  
bPostCode CHAR(4),  
bState VARCHAR(3),  
numberEmployees INTEGER  
);
```

**Publisher** (publisherNo, pName, pStreetNo, pStreetName, pPostCode, pState)

```
> CREATE table Publisher (  
publisherNo CHAR(3) Primary key CHECK ( publisherNo LIKE '[a-zA-Z][0-9][0-9]'),  
pName TEXT,  
pStreetNo TEXT,  
pStreetName TEXT,  
pPostCode CHAR(4),  
pState VARCHAR(3)  
);
```

**Author** (authorID, aFirstName, aLastName)

```
>CREATE table Author (  
authorID CHAR(4) Primary key,  
aFirstName TEXT,  
aLastName TEXT  
);
```

**Book (ISBN, title, publisherNo, genre, retailPrice, paperback)**

```
>CREATE table Book (  
ISBN CHAR(10) Primary key,  
title TEXT NOT NULL,  
publisherNo CHAR(3) CHECK ( publisherNo LIKE '[a-zA-Z][0-9][0-9]'),  
genre TEXT,  
retailPrice INTEGER,  
paperback TEXT,  
FOREIGN KEY (publisherNo) REFERENCES Publisher (publisherNo)  
);
```

**Wrote (ISBN, authorID)**

```
>CREATE table Wrote (  
ISBN CHAR(10),  
authorID CHAR(4),  
PRIMARY KEY (ISBN, authorID),  
FOREIGN KEY (ISBN) REFERENCES Book (ISBN),  
FOREIGN KEY (authorID) REFERENCES Author (authorID)  
);
```

**Inventory (ISBN, branchNo, quantityInStock)**

```
>CREATE table Inventory (  
ISBN CHAR(10),  
branchNo CHAR(3),  
quantityInStock INTERGER,  
PRIMARY KEY (ISBN, branchNo),  
FOREIGN KEY (ISBN) REFERENCES Book (ISBN),  
FOREIGN KEY (branchNo) REFERENCES Branch (branchNo)
```

);

```
Command Prompt - sqlite3 C X + v
Microsoft Windows [Version 10.0.22621.2283]
(c) Microsoft Corporation. All rights reserved.

C:\Users\JP>cd Desktop\IFQ554 Assignment 2 Folder

C:\Users\JP\Desktop\IFQ554 Assignment 2 Folder>sqlite3 Oktomook
SQLite version 3.43.0 2023-08-24 12:36:59
Enter ".help" for usage hints.
sqlite> PRAGMA foreign_keys = 1;
sqlite> CREATE table Branch ( branchNo CHAR(3) Primary key, bName TEXT, bStreetNo TEXT, bStreetName TEXT, bPostCode CHAR
(4), bState VARCHAR(3), numberEmployees INTEGER );
sqlite> CREATE table Publisher ( publisherNo CHAR(3) Primary key CHECK ( publisherNo LIKE '[a-zA-Z][0-9][0-9]'), pName T
EXT, pStreetNo TEXT, pStreetName TEXT, pPostCode CHAR(4), pState VARCHAR(3) );
sqlite> CREATE table Author ( authorID CHAR(4) Primary key, aFirstName TEXT, aLastName TEXT );
sqlite> CREATE table Book ( ISBN CHAR(10) Primary key, title TEXT NOT NULL, publisherNo CHAR(3) CHECK ( publisherNo LIKE
'[a-zA-Z][0-9][0-9]'), genre TEXT, retailPrice INTEGER, paperback TEXT, FOREIGN KEY (publisherNo) REFERENCES Publisher
(publisherNo) );
sqlite> CREATE table Wrote ( ISBN CHAR(10), authorID CHAR(4), PRIMARY KEY (ISBN, authorID), FOREIGN KEY (ISBN) REFERENCE
S Book (ISBN), FOREIGN KEY (authorID) REFERENCES Author (authorID) );
sqlite> CREATE table Inventory ( ISBN CHAR(10), branchNo CHAR(3), quantityInStock INTERGER, PRIMARY KEY (ISBN, branchNo)
, FOREIGN KEY (ISBN) REFERENCES Book (ISBN), FOREIGN KEY (branchNo) REFERENCES Branch (branchNo) );
sqlite> |
```

```
Command Prompt - sqlite3 O X Command Prompt - sqlite3 C X + v
Microsoft Windows [Version 10.0.22621.2283]
(c) Microsoft Corporation. All rights reserved.

C:\Users\JP>cd Desktop\IFQ554 Assignment 2 Folder

C:\Users\JP\Desktop\IFQ554 Assignment 2 Folder>sqlite3 Oktomook
SQLite version 3.43.0 2023-08-24 12:36:59
Enter ".help" for usage hints.
sqlite> .tables
Author      Book        Branch      Inventory  Publisher  Wrote
sqlite> |
```

## Task 2:

List the hotelNo, type and price of each room that is a double, self or deluxe with a price of more than \$110.

> SELECT hotelNo, type, price FROM Room WHERE price > 110 AND type IN ('Double', 'Self', 'Deluxe');

```
Microsoft Windows [Version 10.0.22621.2283]
(c) Microsoft Corporation. All rights reserved.

C:\Users\JP>cd Desktop\IFQ554 Assignment 2 Folder

C:\Users\JP\Desktop\IFQ554 Assignment 2 Folder>sqlite3 Hotel.db
SQLite version 3.43.0 2023-08-24 12:36:59
Enter ".help" for usage hints.
sqlite> SELECT hotelNo, type, price FROM Room WHERE price > 110 AND type IN ('Double', 'Self', 'Deluxe');
H1|Self|135
H2|Double|115
H2|Self|135
H2|Deluxe|160
H3|Self|135
H3|Deluxe|145
H4|Self|135
H5|Double|115
H5|Double|115
H6|Double|115
sqlite> |
```

List the hotelNo which have 2 or more double rooms.

> SELECT hotelNo FROM Room WHERE type = 'Double' GROUP BY hotelNo HAVING COUNT(\*) >= 2;

```
Microsoft Windows [Version 10.0.22621.2283]
(c) Microsoft Corporation. All rights reserved.

C:\Users\JP>cd Desktop\IFQ554 Assignment 2 Folder

C:\Users\JP\Desktop\IFQ554 Assignment 2 Folder>sqlite3 Hotel.db
SQLite version 3.43.0 2023-08-24 12:36:59
Enter ".help" for usage hints.
sqlite> SELECT hotelNo FROM Room WHERE type = 'Double' GROUP BY hotelNo HAVING COUNT(*) >= 2;
H4
H5
sqlite> |
```

How many different guests visited the Grosvenor Hotel?

> SELECT COUNT(DISTINCT guestNo) FROM Booking WHERE hotelNo = (SELECT hotelNo FROM Hotel WHERE hotelName = 'Grosvenor Hotel');

```
Microsoft Windows [Version 10.0.22621.2283]
(c) Microsoft Corporation. All rights reserved.

C:\Users\JP>cd Desktop\IFQ554 Assignment 2 Folder

C:\Users\JP\Desktop\IFQ554 Assignment 2 Folder>sqlite3 Hotel.db
SQLite version 3.43.0 2023-08-24 12:36:59
Enter ".help" for usage hints.
sqlite> SELECT COUNT(DISTINCT guestNo) FROM Booking WHERE hotelNo = (SELECT hotelNo FROM Hotel WHERE hotelName = 'Grosve
nor Hotel');
2
sqlite> |
```

**What is the total income from bookings for the Grosvenor Hotel?**

>SELECT SUM(price) FROM Booking b JOIN Room r ON b.roomNo = r.roomNo WHERE b.hotelNo =  
(SELECT hotelNo FROM Hotel WHERE hotelName = 'Grosvenor Hotel');

```
Microsoft Windows [Version 10.0.22621.2283]
(c) Microsoft Corporation. All rights reserved.

C:\Users\JP>cd Desktop\IFQ554 Assignment 2 Folder

C:\Users\JP\Desktop\IFQ554 Assignment 2 Folder>sqlite3 Hotel.db
SQLite version 3.43.0 2023-08-24 12:36:59
Enter ".help" for usage hints.
sqlite> SELECT SUM(price) FROM Booking b JOIN Room r ON b.roomNo = r.roomNo WHERE b.hotelNo = (SELECT hotelNo FROM Hotel
 WHERE hotelName = 'Grosvenor Hotel');
3270
sqlite> |
```

**List all the guests' names who have stayed in a hotel.**

> SELECT DISTINCT g.guestName FROM Guest g JOIN Booking b ON g.guestNo = b.guestNo;

```
Microsoft Windows [Version 10.0.22621.2283]
(c) Microsoft Corporation. All rights reserved.

C:\Users\JP>cd Desktop\IFQ554 Assignment 2 Folder

C:\Users\JP\Desktop\IFQ554 Assignment 2 Folder>sqlite3 Hotel.db
SQLite version 3.43.0 2023-08-24 12:36:59
Enter ".help" for usage hints.
sqlite> SELECT DISTINCT g.guestName FROM Guest g JOIN Booking b ON g.guestNo = b.guestNo;
Mark Walpole
Robert Chen
Esther Lin
Harry Sarin
sqlite> |
```

### Task 3:

**Step 1:** Write commands to insert one row of data in each of the Hotel database tables.

#### Booking Table

> INSERT INTO Booking (hotelNo, guestNo, dateFrom, dateTo, roomNo) VALUES ('H9', 'G9', '2023-09-13', '2023-09-14', 'R1');

```
Command Prompt - sqlite3 h...
C:\Users\JP>cd C:\Users\JP\Desktop\IFQ554 Assignment 2 Folder

C:\Users\JP\Desktop\IFQ554 Assignment 2 Folder>sqlite3 Hotel.db
SQLite version 3.43.0 2023-08-24 12:36:59
Enter ".help" for usage hints.
sqlite> INSERT INTO Booking (hotelNo, guestNo, dateFrom, dateTo, roomNo) VALUES ('H9', 'G9', '2023-09-13', '2023-09-14', 'R1');
sqlite> SELECT * FROM Booking
...>
H1|G1|2018-03-02|2018-03-25|R1
H2|G3|2018-05-25|2018-05-28|R1
H3|G1|2018-06-20|2018-06-28|R1
H4|G2|2018-05-25|2018-05-28|R1
H5|G4|2018-05-25|2018-05-27|R1
H1|G2|2018-05-27|2018-05-27|R1
H2|G3|2018-03-19|2018-03-22|R1
H2|G1|2018-06-07|2018-06-18|R1
H4|G2|2018-04-25|2018-04-28|R1
H5|G4|2018-05-15|2018-05-17|R1
H1|G1|2019-03-02|2019-03-25|R1
H1|G2|2019-03-02|2019-03-25|R2
H6|G1|2019-03-02|2019-03-25|R1
H7|G3|2020-03-20|2020-03-26|R2
H7|G1|2019-03-02|2019-03-24|R3
H7|G1|2019-02-02|2019-02-15|R3
H4|G1|2019-02-02|2019-02-15|R3
H1|G1|2019-04-29|2019-05-12|R1
H9|G9|2023-09-13|2023-09-14|R1
sqlite> |
```

#### Guest Table

> INSERT INTO Guest (guestNo, guestName, guestAddress) VALUES ('G6', 'Joshua Parkes', 'Gympie');

```
Command Prompt - sqlite3 x Command Prompt x Command Prompt x Command Prompt x + - □ x
Microsoft Windows [Version 10.0.22621.2283]
(c) Microsoft Corporation. All rights reserved.

C:\Users\JP>cd C:\Users\JP\Desktop\IFQ554 Assignment 2 Folder

C:\Users\JP\Desktop\IFQ554 Assignment 2 Folder>sqlite3 Hotel.db
SQLite version 3.43.0 2023-08-24 12:36:59
Enter ".help" for usage hints.
sqlite> INSERT INTO Guest (guestNo, guestName, guestAddress) VALUES ('G6', 'Joshua Parkes', 'Gympie');
sqlite> SELECT * FROM Guest;
G1|Mark Walpole|Brisbane
G2|Robert Chen|Shanghai
G3|Esther Lin|Taipei
G4|Harry Sarin|Mumbai
G5|John English|Edinburgh
G6|Joshua Parkes|Gympie
sqlite> |
```

## Hotel Table

> INSERT INTO Hotel (hotelNo, hotelName, city) VALUES ('H9', 'Database Hotel', 'Brisbane');

```
Command Prompt - sqlite3 | x Command Prompt - sqlite3 x Command Prompt x Command Prompt x + - □ x
Microsoft Windows [Version 10.0.22621.2283]
(c) Microsoft Corporation. All rights reserved.

C:\Users\JP>cd C:\Users\JP\Desktop\IFQ554 Assignment 2 Folder

C:\Users\JP\Desktop\IFQ554 Assignment 2 Folder>sqlite3 Hotel.db
SQLite version 3.43.0 2023-08-24 12:36:59
Enter ".help" for usage hints.
sqlite> INSERT INTO Hotel (hotelNo, hotelName, city) VALUES ('H9', 'Database Hotel', 'Brisbane');
sqlite> SELECT * FROM Hotel;
H1|Grosvenor Hotel|Cairns
H2|Ridge Hotel|Townsville
H3|Plaza Hotel|Mackay
H4|Backpacker Hotel|Brisbane
H5|Rocky Hotel|Rockhampton
H6|Abbey Hotel|Cairns
H7|Coconut Hotel|Cairns
H9|Database Hotel|Brisbane
sqlite> |
```

## Room Table

> INSERT INTO Room (roomNo, hotelNo, type, price) VALUES ('R1', 'H9', 'Deluxe', '150');

```
Command Prompt - sqllite3 H x Command Prompt - sqllite3 H x Command Prompt - sqllite3 H x Command Prompt
C:\Users\JP>cd C:\Users\JP\Desktop\IFQ554 Assignment 2 Folder
C:\Users\JP\Desktop\IFQ554 Assignment 2 Folder>sqlite3 Hotel.db
SQLite version 3.43.0 2023-08-24 12:36:59
Enter ".help" for usage hints.
sqlite> INSERT INTO Room (roomNo, hotelNo, type, price) VALUES ('R1', 'H9', 'Deluxe', '150');
sqlite> SELECT * FROM Room;
R1|H1|Single|90
R2|H1|Single|90
R3|H1|Double|105
R4|H1|Self|135
R5|H1|Single|95
R1|H2|Single|95
R2|H2|Single|90
R3|H2|Double|115
R4|H2|Self|135
R5|H2|Deluxe|160
R1|H3|Single|90
R2|H3|Single|90
R3|H3|Double|105
R4|H3|Self|135
R5|H3|Deluxe|145
R1|H4|Single|90
R2|H4|Single|90
R3|H4|Double|105
R4|H4|Self|135
R5|H4|Double|105
R1|H5|Single|90
R2|H5|Single|90
R3|H5|Double|105
R4|H5|Double|115
R5|H5|Double|115
R2|H6|Single|90
R1|H6|Double|115
R1|H7|Single|90
R2|H7|Single|90
R3|H7|Double|105
R1|H9|Deluxe|150
sqlite> |
```

**Step 2:** Write a command to delete the row you inserted in the table Guest.

>DELETE FROM Guest WHERE guestNo = 'G6';

```
Command Prompt - sqllite3 H x Command Prompt - sqllite3 H x + v
Microsoft Windows [Version 10.0.22621.2283]
(c) Microsoft Corporation. All rights reserved.

C:\Users\JP>cd Desktop\IFQ554 Assignment 2 Folder
C:\Users\JP\Desktop\IFQ554 Assignment 2 Folder>sqlite3 Hotel.db
SQLite version 3.43.0 2023-08-24 12:36:59
Enter ".help" for usage hints.
sqlite> SELECT * FROM Guest;
G1|Mark Walpole|Brisbane
G2|Robert Chen|Shanghai
G3|Esther Lin|Taipei
G4|Harry Sarin|Mumbai
G5|John English|Edinburgh
G6|Joshua Parkes|Gympie
sqlite> DELETE FROM Guest WHERE guestNo = 'G6';
sqlite> SELECT * FROM Guest;
G1|Mark Walpole|Brisbane
G2|Robert Chen|Shanghai
G3|Esther Lin|Taipei
G4|Harry Sarin|Mumbai
G5|John English|Edinburgh
sqlite> |
```

**Step 3:** Write a command to update the price of all rooms by 15%.

> UPDATE Room SET price = price\*1.15;



```
Command Prompt - sqllite3 H x Command Prompt - sqllite3 H x + v
sqlite> UPDATE Room SET price = price*1.15;
sqlite> SELECT * FROM Room;
R1|H1|Single|103.5
R2|H1|Single|103.5
R3|H1|Double|120.75
R4|H1|Self|155.25
R5|H1|Single|109.25
R1|H2|Single|109.25
R2|H2|Single|103.5
R3|H2|Double|132.25
R4|H2|Self|155.25
R5|H2|Deluxe|184
R1|H3|Single|103.5
R2|H3|Single|103.5
R3|H3|Double|120.75
R4|H3|Self|155.25
R5|H3|Deluxe|166.75
R1|H4|Single|103.5
R2|H4|Single|103.5
R3|H4|Double|120.75
R4|H4|Self|155.25
R5|H4|Double|120.75
R1|H5|Single|103.5
R2|H5|Single|103.5
R3|H5|Double|120.75
R4|H5|Double|132.25
R5|H5|Double|132.25
R2|H6|Single|103.5
R1|H6|Double|132.25
R1|H7|Single|103.5
```

## Task 4:

Write a command to create an index on guestName of the Guest table.

> CREATE INDEX name\_of\_guest ON Guest(guestName);

> PRAGMA index\_list('Guest');

```
Command Prompt - sqllite3 H x Command Prompt - sqllite3 H x + v
Microsoft Windows [Version 10.0.22621.2283]
(c) Microsoft Corporation. All rights reserved.

C:\Users\JP>cd Desktop\IFQ554 Assignment 2 Folder

C:\Users\JP\Desktop\IFQ554 Assignment 2 Folder>sqlite3 Hotel.db
SQLite version 3.43.0 2023-08-24 12:36:59
Enter ".help" for usage hints.
sqlite> CREATE INDEX name_of_guest ON Guest(guestName);
sqlite> PRAGMA index_list('Guest');
0|name_of_guest|0|c|0
1|sqlite_autoindex_Guest_1|1|pk|0
sqlite>
```

Write a command to create a view to list the information (hotelName, roomType and the total number of rooms booked) of the hotels which are in Cairns.

>CREATE VIEW taskfour\_view AS SELECT h.hotelName, r.type AS roomType, COUNT (b.roomNo) AS total  
FROM Hotel h JOIN Room r ON h.hotelNo = r.hotelNo JOIN Booking b ON r.roomNo = b.roomNo WHERE  
h.city = 'Cairns' GROUP BY h.hotelName, r.type;

```
Command Prompt - sqllite3 H x Command Prompt - sqllite3 H x + v
Microsoft Windows [Version 10.0.22621.2283]
(c) Microsoft Corporation. All rights reserved.

C:\Users\JP>cd Desktop\IFQ554 Assignment 2 Folder

C:\Users\JP\Desktop\IFQ554 Assignment 2 Folder>sqlite3 Hotel.db
SQLite version 3.43.0 2023-08-24 12:36:59
Enter ".help" for usage hints.
sqlite> CREATE VIEW taskfour_view AS SELECT h.hotelName, r.type AS roomType, COUNT (b.roomNo) AS total FROM Hotel h JOIN
Room r ON h.hotelNo = r.hotelNo JOIN Booking b ON r.roomNo = b.roomNo WHERE h.city = 'Cairns' GROUP BY h.hotelName, r.t
ype;
sqlite> SELECT * FROM taskfour_view;
Abbey Hotel|Double|14
Abbey Hotel|Single|2
Coconut Hotel|Double|3
Coconut Hotel|Single|16
Grosvenor Hotel|Double|3
Grosvenor Hotel|Single|16
sqlite> |
```

## Task 5:

User Nikki must be able to add records to the Booking table:

>GRANT INSERT

On Booking

To nikki;

User Nikki must be able to remove records from the Booking table:

>GRANT DELETE

On Booking

To nikki;

User Phil is no longer allowed to add data to the Guest table

>REVOKE INSERT

On Guest

To phil;

User Phil is no longer allowed to delete records from the Guest table

>REVOKE DELETE

On Guest

To phil;

## **Task 6:**

During the COVID19 pandemic the Queensland Government required mandatory evidence of infection for some jobs and individuals. This was done through reporting of rapid antigen test (RAT) data. This raises ethical concerns for data usage, some that are unique due to the mandatory requirements of the Government.

Concerns for the collection of data need to focus on how the data should stay anonymous as possible to prevent the data from becoming personally identifiable information and a privacy violation. User consent is an ethical concern in collecting data. In the case of RAT, data subjects have limited consent powers of their data. If testing is forced upon them by the government, the user has limited agency in self-determination of their data. However, in such a circumstance they should have the power and right to opt-in or out of any further secondary use of their data aside from the primary goal of proving their COVID infection status to the immediate authorities.

Full transparency of data collection and use is a key ethical concern, especially in the case where collection of data is mandatory. Data subjects should have their concerns and questions answered if requested into what data is taken, what format, who has access to and what actually is done with it. This gives data subjects information to inform their consent options and also provides trust with various stakeholders. This can only improve data management processes and data governance.

Successful data management is essentially the achievement and practical application of the goals and objectives of good data governance. Ethical considerations and codes of ethics should be an integral part of data management, however, to be effective it also requires surrounding and supporting solutions such as legal safeguards, regulatory compliances and data management policy at institutional levels. Data management is improved when all stakeholders have some agreement and streamlining of their goals, objectives and policies in regard to data governance.

Generally, user consent and the associated privacy considerations of a personal data subject involves both rights and responsibilities (for example accepting user agreements to access services). However, when data collection is mandatory or forced upon the data subject such as RAT, this brings the full onus of responsibility onto those managing the data and those that are forcing the user to provide that information. This further highlights the need for strong data governance that fosters and facilitates successful data management, which also includes transparency for the data subject. Important lessons for the future can be learned from the pandemic and the experiences such as government collection of RAT data.

As traditional services move to digital platforms and delivery, mandatory data collection will become more common. An example of this is the Australian banking system in which physical bank tellers are closing at a rapid rate indicating acceleration towards a cashless society. These developments lead to

customers with no choice but to give more of their data to financial institutions just to access basic services. This shows the necessity for continuous improvement and implementation of good data governance.

## Task 7:

### Invoice

invoiceID	customerID
867547	C001
867547	C001
867547	C001
867548	C004
867549	C004

### Product

productID	productDescription	vendorID
RS-E3422QW	Rotary Sander	V211
DB-300932X	0.25-in. drill bit	V211
BS-995748G	Band Saw	V309
RS-E3422QW	Rotary Sander	V211
PD-778345P	Power Drill	V157

### Sold

invoiceID	productID	numberSold	productPrice
867547	RS-E3422QW	10	\$60.00
867547	DB-300932X	82	\$9.00
867547	BS-995748G	13	\$55.00
867548	RS-E3422QW	23	\$50.00
867549	PD-778345P	15	\$90.00

### Functional Dependencies:

invoiceID ---> customerID

productID ---> productDescription, vendorID

invoiceID, productID ---> numberSold, productPrice

