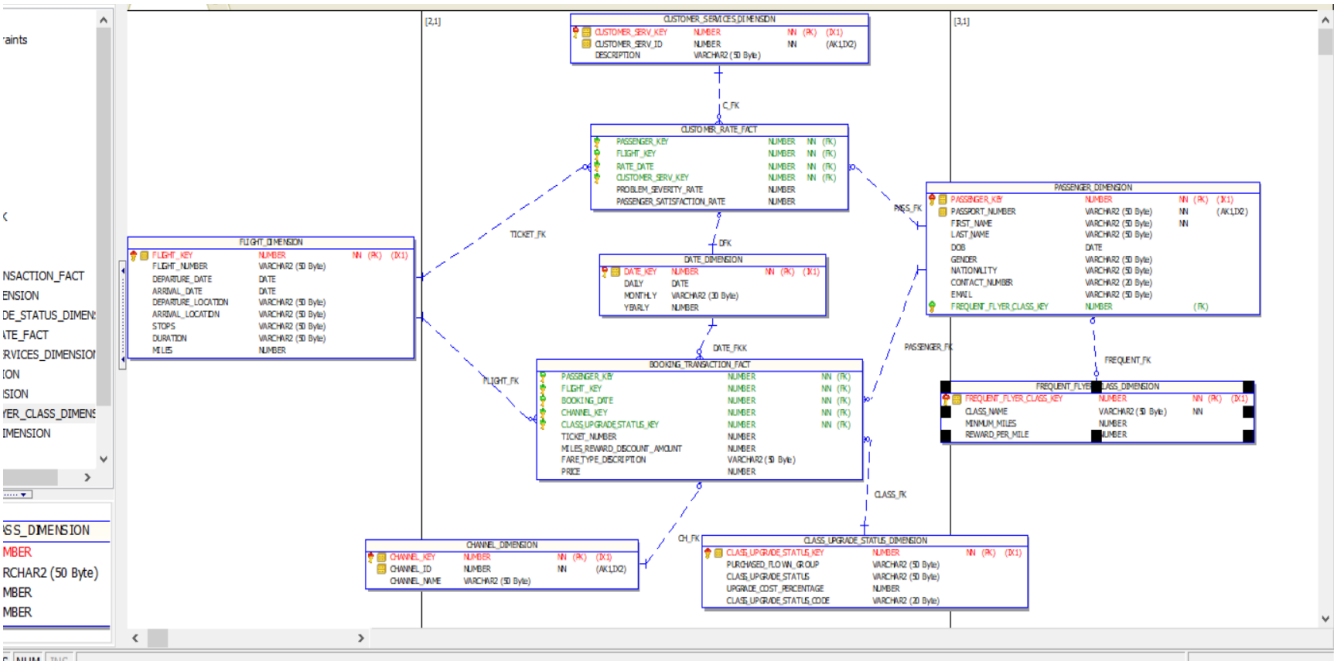
3.Pysical Model

Type of indexes in the physical design model

1-primary key index: Each dimension must have a pk

2-Foreign key index: Each fact table has foreign keys that connect to its dimensions



4.DDLS

PASSENGER_DIMENSION

```
CREATE TABLE PASSENGER_DIMENSION
(
  PASSENGER_KEY            NUMBER,
  PASSPORT_NUMBER          VARCHAR2(50 BYTE)   NOT NULL,
  FIRST_NAME               VARCHAR2(50 BYTE)   NOT NULL,
  LAST_NAME                VARCHAR2(50 BYTE),
  DOB                      DATE,
  GENDER                   VARCHAR2(50 BYTE),
  NATIONALITY              VARCHAR2(50 BYTE),
  CONTACT_NUMBER           VARCHAR2(20 BYTE),
  EMAIL                    VARCHAR2(50 BYTE),
  FREQUENT_FLYER_CLASS_KEY  NUMBER
)
```

```
ALTER TABLE PASSENGER_DIMENSION ADD (
  CONSTRAINT PASSENGER_DIMENSION_PK
 PRIMARY KEY (PASSENGER_KEY)
);

ALTER TABLE PASSENGER_DIMENSION ADD (
  CONSTRAINT FREQUENT_FK
 FOREIGN KEY (FREQUENT_FLYER_CLASS_KEY)
 REFERENCES HR.FREQUENT_FLYER_CLASS_DIMENSION
(FREQUENT_FLYER_CLASS_KEY));
```

PASSENGER_KEY	PASSPORT_NUMBER	FIRST_NAME	LAST_NAME	DOB	GENDER	NATIONALITY	CONTACT_NUMBER	EMAIL	FREQUENT_FLYER_CLASS_KEY
9	A45789645	Ahmed	Serry	01/01/1996	MALE	EGYPTIAN	01144556678	ahmed@gmail.com	5
1	A13198480	OMAR	LOAY	01/01/1969	MALE	EGYPTIAN	01144556699	omar@gmail.com	1
2	A12345678	Nayra	Gabr	05/01/1967	Female	EGYPTIAN	01144598678	Nayra@gmail.com	4
4	A22446688	Mariem	Shawky	05/05/1996	Female	EGYPTIAN	01144223189	Mariem@gmail.com	4
5	A24681045	Mayar	Hany	05/09/1997	Female	EGYPTIAN	01144897816	Mayar@gmail.com	2
6	A45324168	Menna	Mohamed	20/11/1999	Female	EGYPTIAN	01144897818	Menna@gmail.com	5
7	A45524168	Mena	Mohamed	25/01/1999	Female	EGYPTIAN	01144897817	Mena@gmail.com	5
8	A12345689	Ranem	Wael	01/01/1999	Female	EGYPTIAN	01112345678	ranem@gmail.com	5

FREQUENT_FLYER_CLASS_DIMENSION

```
CREATE TABLE FREQUENT_FLYER_CLASS_DIMENSION
(
  FREQUENT_FLYER_CLASS_KEY  NUMBER,
  CLASS_NAME              VARCHAR2(50 BYTE)   NOT NULL,
  MINMUM_MILES             NUMBER,
  REWARD_PER_MILE          NUMBER
)
ALTER TABLE FREQUENT_FLYER_CLASS_DIMENSION ADD (
  CONSTRAINT FREQUENT_FLYER_CLASS_DIMENS_PK
 PRIMARY KE (FREQUENT_FLYER_CLASS_KEY) );
```

Read Only Auto Refresh

FREQUENT_FLYER_CLASS_KEY	CLASS_NAME	MINMUM_MILES	REWARD_PER_MILE
1	SILIVER	1000	0.2
2	GOLD	3000	0.5
3	PLATINUM	5000	0.6
4	TITANIUM	8000	0.7
5	NON FF	0	0

FLIGHT_DIMENSION

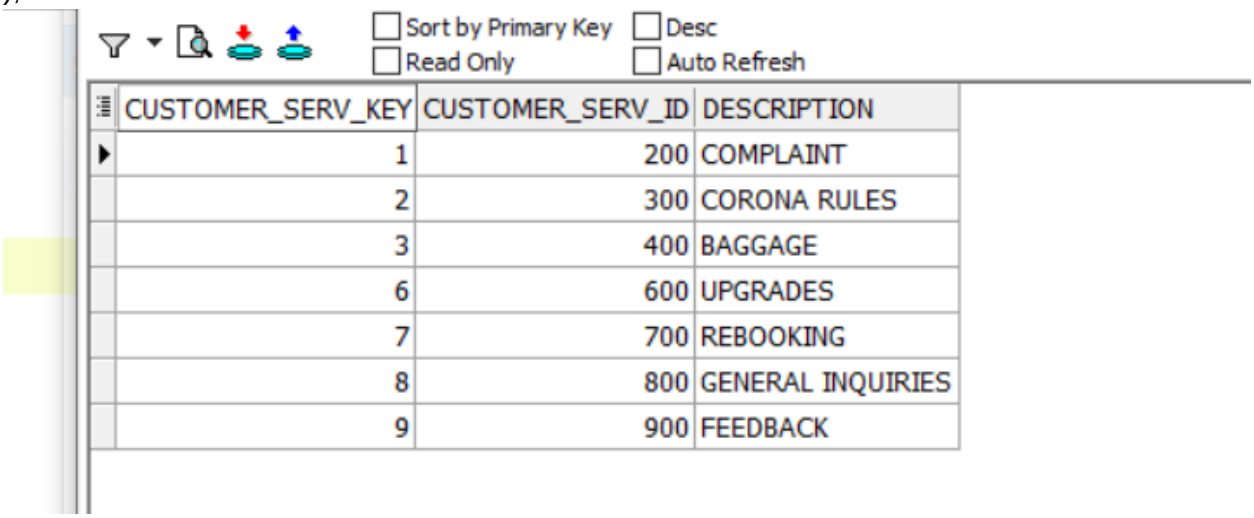
```
CREATE TABLE FLIGHT_DIMENSION
(
  FLIGHT_KEY          NUMBER,
  FLIGHT_NUMBER       VARCHAR2(50 BYTE),
  DEPARTURE_DATE      DATE,
  ARRIVAL_DATE        DATE,
  DEPARTURE_LOCATION  VARCHAR2(50 BYTE),
  ARRIVAL_LOCATION    VARCHAR2(50 BYTE),
  STOPS               VARCHAR2(50 BYTE),
  DURATION            VARCHAR2(50 BYTE),
  MILES               NUMBER
)
ALTER TABLE FLIGHT_DIMENSION ADD (
  CONSTRAINT TICKET_DIMENSION_PK
 PRIMARY KEY
 (FLIGHT_KEY)
);
```

FLIGHT_KEY	FLIGHT_NUMBER	DEPARTURE_DATE	ARRIVAL_DATE	DEPARTURE_LOCATION	ARRIVAL_LOCATION	STOPS	DURATION	MILES
1	AM123	10/01/2021	11/01/2021	CAIRO	SAN FRANCISCO	2 STOPS	23	7450
2	AH435	22/11/2021	22/11/2021	CAIRO	FRANKURT	DIRECT	5	3000
3	OR569	23/12/2021	25/12/2021	CAIRO	REYKJAVIK	1 STOP	9	3200
4	YM445	01/02/2022	02/02/2022	CAIRO	NEW YORK	DIRECT	10	5600
5	NY123	26/01/2022	27/01/2022	ALEX	ISTANBUL	DIRECT	4	1740
6	UM145	27/01/2022	27/01/2022	ALEX	LONDON	1 STOP	6	3055

CUSTOMER_SERVICES_DIMENSION

```
CREATE TABLE CUSTOMER_SERVICES_DIMENSION
(
  CUSTOMER_SERV_KEY  NUMBER,
  CUSTOMER_SERV_ID   NUMBER                     NOT NULL,
  DESCRIPTION        VARCHAR2(50 BYTE)
)

ALTER TABLE CUSTOMER_SERVICES_DIMENSION ADD (
  CONSTRAINT CUSTOMER_SERVICES_PK
  PRIMARY KEY  (CUSTOMER_SERV_KEY)

CONSTRAINT CUS_SERV_UN
 UNIQUE (CUSTOMER_SERV_ID)
```

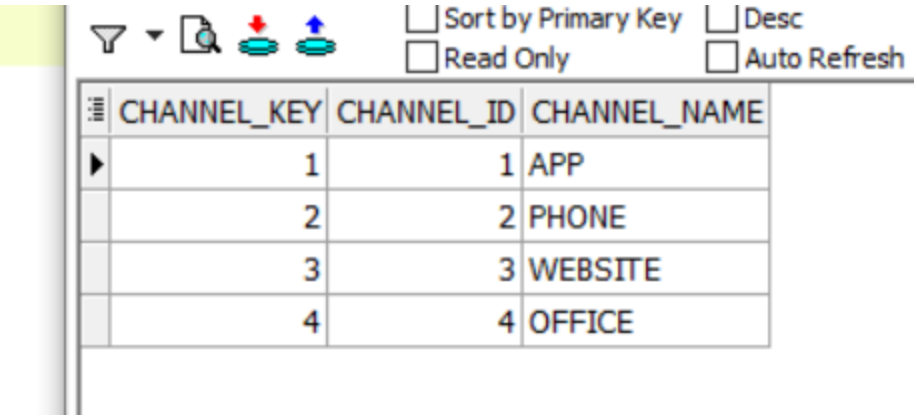
);



CUSTOMER_SERV_KEY	CUSTOMER_SERV_ID	DESCRIPTION
1	200	COMPLAINT
2	300	CORONA RULES
3	400	BAGGAGE
6	600	UPGRADES
7	700	REBOOKING
8	800	GENERAL INQUIRIES
9	900	FEEDBACK

CHANNEL_DIMENSION

```
CREATE TABLE CHANNEL_DIMENSION
(
  CHANNEL_KEY   NUMBER,
  CHANNEL_ID    NUMBER                      NOT NULL,
  CHANNEL_NAME  VARCHAR2(50 BYTE)
)

ALTER TABLE CHANNEL_DIMENSION ADD (
  CONSTRAINT CHANNEL_DIMENSION_PK
 PRIMARY KEY
 (CHANNEL_KEY)
 CONSTRAINT CHANNEL_UN
 UNIQUE (CHANNEL_ID) );
```



CHANNEL_KEY	CHANNEL_ID	CHANNEL_NAME
1	1	APP
2	2	PHONE
3	3	WEBSITE
4	4	OFFICE

CLASS_UPGRADE_STATUS_DIMENSION

```
CREATE TABLE CLASS_UPGRADE_STATUS_DIMENSION
(
  CLASS_UPGRADE_STATUS_KEY   NUMBER,
  PURCHASED_FLOWN_GROUP      VARCHAR2(50 BYTE),
  CLASS_UPGRADE_STATUS       VARCHAR2(50 BYTE),
  UPGRADE_COST_PERCENTAGE    NUMBER,
  CLASS_UPGRADE_STATUS_CODE  VARCHAR2(20 BYTE)
)
ALTER TABLE CLASS_UPGRADE_STATUS_DIMENSION ADD (
  CONSTRAINT CLASS_UPDATE_DIMENSION_PK
 PRIMARY KEY
 (CLASS_UPGRADE_STATUS_KEY));
```

CLASS_UPGRADE_STATUS_KEY	PURCHASED_FLOWN_GROUP	CLASS_UPGRADE_STATUS	UPGRADE_COST_PERCENTAGE	CLASS_UPGRADE_STATUS_CODE
1	ECONOMY-ECONOMY	NO CLASS UPGRADE		EE
2	ECONOMY-BUSINESS	UPGRADE	15	EB
3	ECONOMY-FIRST	UPGRADE	35	EF
4	BUSINESS-BUSINESS	NO CLASS UPGRADE		BB
5	BUSINESS-FIRST	UPGRADE	30	BF
6	FIRST-FIRST	NO CLASS UPGRADE		FF

DATE_DIMENSION

```
CREATE TABLE DATE_DIMENSION
(
  DATE_KEY  NUMBER,
  DAILY     DATE,
  MONTHLY   VARCHAR2(30 BYTE),
  YEARLY    NUMBER
)
ALTER TABLE DATE_DIMENSION ADD (
```

```
 CONSTRAINT DATE_DIMENSION_PK
PRIMARY KEY(DATE_KEY));
```

DATE_KEY	DAILY	MONTHLY	YEARLY
1	01/01/2021	JAN	2021
2	11/01/2021	JAN	2021
3	20/11/2021	NOV	2021
4	22/11/2021	NOV	2021
5	23/12/2021	DEC	2021
6	25/12/2021	DEC	2021
7	01/01/2022	JAN	2022
8	25/01/2022	JAN	2022
9	26/01/2022	JAN	2022
10	27/01/2022	JAN	2022
11	28/01/2022	JAN	2022
12	01/02/2022	FEB	2022
13	02/02/2022	FEB	2022
14	09/01/2021	JAN	2021
15	01/10/2021	OCT	2021
16	02/10/2021	OCT	2021
17	03/10/2021	OCT	2021
18	01/11/2021	NOV	2021
19	01/12/2021	DEC	2021

CUSTOMER_RATE_FACT

```
CREATE TABLE CUSTOMER_RATE_FACT
(
  PASSENGER_KEY              NUMBER          NOT NULL,
  FLIGHT_KEY                 NUMBER          NOT NULL,
  RATE_DATE                  NUMBER          NOT NULL,
  CUSTOMER_SERV_KEY          NUMBER          NOT NULL,
  PROBLEM_SEVERITY_RATE      NUMBER,
  PASSENGER_SATISFACTION_RATE  NUMBER
)
ALTER TABLE CUSTOMER_RATE_FACT ADD (
  CONSTRAINT C_FK
 FOREIGN KEY (CUSTOMER_SERV_KEY)
 REFERENCES CUSTOMER_SERVICES_DIMENSION (CUSTOMER_SERV_KEY),
  CONSTRAINT DFK
```

```
FOREIGN KEY (RATE_DATE)
REFERENCES DATE_DIMENSION (DATE_KEY),
 CONSTRAINT PASS_FK
FOREIGN KEY (PASSENGER_KEY)
REFERENCES PASSENGER_DIMENSION (PASSENGER_KEY),
 CONSTRAINT TICKET_FK
FOREIGN KEY (FLIGHT_KEY)
REFERENCES FLIGHT_DIMENSION (FLIGHT_KEY));
```

Read Only Auto Refresh

PASSENGER_KEY	FLIGHT_KEY	RATE_DATE	CUSTOMER_SERV_KEY	PROBLEM_SEVERITY_RATE	PASSENGER_SATISFACTION_RATE
1	1	3	3	10	2
1	1	1	2	3	9
1	2	3	8	1	10
1	4	8	2	3	10
2	1	3	3	10	2
2	4	10	7	2	2
4	1	3	3	10	2
4	6	11	8	4	7
5	2	2	2	3	10
5	6	6	6	5	8
6	3	2	2	4	10
7	3	6	2	5	9
1	1	2	9	10	1
2	1	2	9	10	1
4	1	2	9	10	1
1	2	5	9	1	10
4	2	5	9	1	10
5	2	5	9	2	9
4	3	9	9	3	7
6	3	9	9	2	10
7	3	10	9	1	10
9	5	10	9	1	9
1	5	10	9	1	9

BOOKING_TRANSACTION_FACT

```
CREATE TABLE BOOKING_TRANSACTION_FACT
(
  PASSENGER_KEY              NUMBER         NOT NULL,
  FLIGHT_KEY                 NUMBER         NOT NULL,
  BOOKING_DATE               NUMBER         NOT NULL,
  CHANNEL_KEY                NUMBER         NOT NULL,
  CLASS_UPGRADE_STATUS_KEY   NUMBER         NOT NULL,
  TICKET_NUMBER              NUMBER,
```

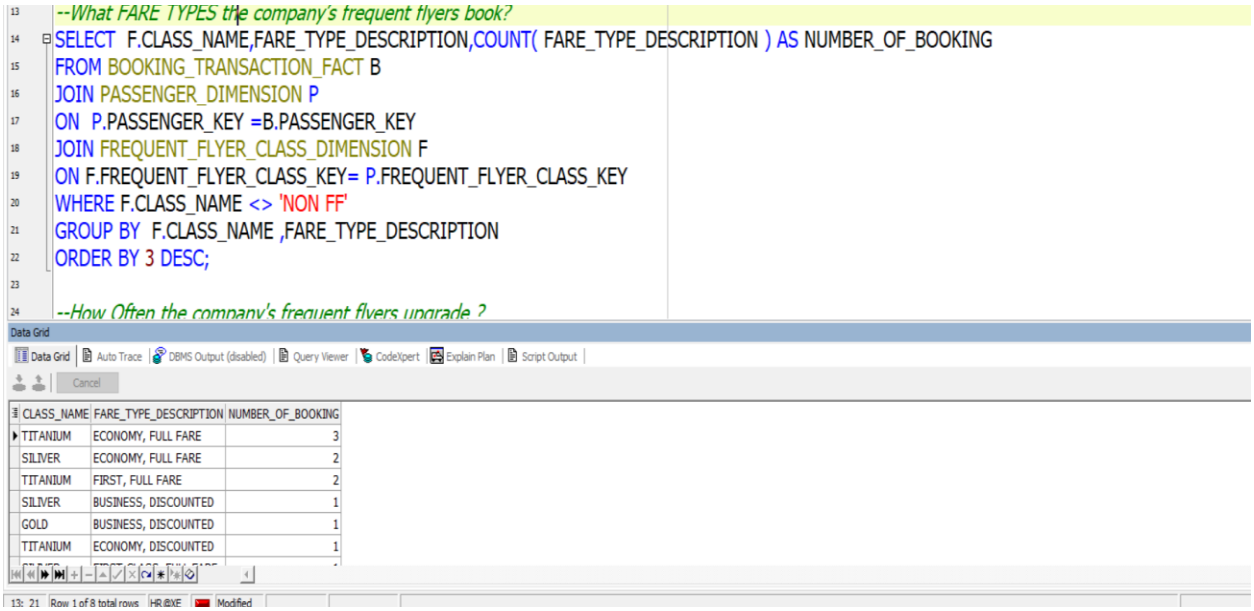
```
  MILES_REWARD_DISCOUNT_AMOUNT  NUMBER,
  FARE_TYPE_DESCRIPTION         VARCHAR2(50 BYTE),
  PRICE                         NUMBER
)

ALTER TABLE BOOKING_TRANSACTION_FACT ADD (
  CONSTRAINT CH_FK
 FOREIGN KEY (CHANNEL_KEY)
 REFERENCES CHANNEL_DIMENSION (CHANNEL_KEY),
  CONSTRAINT CLASS_FK
 FOREIGN KEY (CLASS_UPGRADE_STATUS_KEY)
 REFERENCES CLASS_UPGRADE_STATUS_DIMENSION
(CLASS_UPGRADE_STATUS_KEY),
  CONSTRAINT DATE_FKK
 FOREIGN KEY (BOOKING_DATE)
 REFERENCES DATE_DIMENSION (DATE_KEY),
  CONSTRAINT FLIGHT_FK
 FOREIGN KEY (FLIGHT_KEY)
 REFERENCES FLIGHT_DIMENSION (FLIGHT_KEY),
  CONSTRAINT PASSENGER_FK
 FOREIGN KEY (PASSENGER_KEY)
 REFERENCES PASSENGER_DIMENSION (PASSENGER_KEY));
```

PASSENGER_KEY	FLIGHT_KEY	BOOKING_DATE	CHANNEL_KEY	CLASS_UPGRADE_STATUS_KEY	TICKET_NUMBER	MILES_REWARD_DISCOUNT_AMOUNT	FARE_TYPE_DESCRIPTION	PRICE
2	1	1	3	2	2	0	ECONOMY, FULL FARE	25000
4	1	1	1	1	3	0	ECONOMY, FULL FARE	25000
1	2	3	1	1	4	0	ECONOMY, FULL FARE	10000
5	2	2	1	4	5	0	BUSINESS, DISCOUNTED	12000
4	2	2	1	1	6	4470	ECONOMY, DISCOUNTED	8000
4	3	5	2	1	7	0	ECONOMY, FULL FARE	18000
6	3	1	1	2	8	0	ECONOMY, DISCOUNTED	15000
7	3	2	4	1	9	0	ECONOMY, FULL FARE	18000
1	4	8	1	1	10	7315	ECONOMY, FULL FARE	23000
2	4	9	4	6	11	0	FIRST, FULL FARE	60000
8	4	2	4	1	12	0	ECONOMY, FULL FARE	23000
9	5	7	3	1	13	0	ECONOMY, DISCOUNTED	5000
1	5	7	3	4	14	3360	BUSINESS, DISCOUNTED	9500
4	6	10	2	6	15	0	FIRST, FULL FARE	25000
1	1	1	1	6	1	0	FIRST CLASS, FULL FARE	50000
5	6	4	1	2	16	1500	ECONOMY, DISCOUNTED	7000

5. SQL Queries

1-What FARE TYPES the company's frequent flyers book?

```
SELECT  F.CLASS_NAME,FARE_TYPE_DESCRIPTION,COUNT(
FARE_TYPE_DESCRIPTION ) AS NUMBER_OF_BOOKING
FROM BOOKING_TRANSACTION_FACT B
JOIN PASSENGER_DIMENSION P
ON  P.PASSENGER_KEY =B.PASSENGER_KEY
JOIN FREQUENT_FLYER_CLASS_DIMENSION F
ON F.FREQUENT_FLYER_CLASS_KEY= P.FREQUENT_FLYER_CLASS_KEY
WHERE F.CLASS_NAME <> 'NON FF'
GROUP BY  F.CLASS_NAME ,FARE_TYPE_DESCRIPTION
ORDER BY 3 DESC;
```



2- ORDER channels that make most of the reservation processes ?

```
SELECT COUNT(B.PASSENGER_KEY) AS TOTAL_BOOKING ,C.CHANNEL_NAME
FROM
BOOKING_TRANSACTION_FACT B , CHANNEL_DIMENSION C
WHERE B.CHANNEL_KEY=C.CHANNEL_KEY
GROUP BY C.CHANNEL_NAME
ORDER BY TOTAL_BOOKING DESC ;
```

```
-- ORDER channels that make most the reservation processes ?
SELECT COUNT(B.PASSENGER_KEY) AS TOTAL_BOOKING ,C.CHANNEL_NAME
FROM
BOOKING_TRANSACTION_FACT B , CHANNEL_DIMENSION C
WHERE B.CHANNEL_KEY=C.CHANNEL_KEY
GROUP BY C.CHANNEL_NAME
ORDER BY TOTAL_BOOKING DESC ;
```

Data Grid | Auto Trace | DBMS Output (disabled) | Query Viewer | CodeXpert | Explain Plan | Script Output

Cancel

TOTAL_BOOKING	CHANNEL_NAME
8	APP
3	OFFICE
3	WEBSITE
2	PHONE

3-List flights that have the worst Customer Rate ?

```
SELECT COUNT(C.PASSENGER_SATISFACTION_RATE)  AS
NUMBER_OF_NEGATIVE_RATE ,F.FLIGHT_NUMBER
FROM
CUSTOMER_RATE_FACT  C
JOIN FLIGHT_DIMENSION F
ON
C.FLIGHT_KEY=F.FLIGHT_KEY
WHERE C.PASSENGER_SATISFACTION_RATE <= 3
GROUP BY F.FLIGHT_NUMBER
ORDER BY NUMBER_OF_NEGATIVE_RATE DESC;
```

```sql
--List flights that have worst  Customer Rate?
SELECT COUNT(C.PASSENGER_SATISFACTION_RATE)  AS NUMBER_OF_NEGATIVE_RATE ,F.FLIGHT_NUMBER
FROM
CUSTOMER_RATE_FACT  C
JOIN FLIGHT_DIMENSION F
ON
C.FLIGHT_KEY=F.FLIGHT_KEY
WHERE C.PASSENGER_SATISFACTION_RATE <= 3
GROUP BY F.FLIGHT_NUMBER
ORDER BY NUMBER_OF_NEGATIVE_RATE DESC;
```

Data Grid | Auto Trace | DBMS Output (disabled) | Query Viewer | CodeXpert | Explain Plan | Script Output

Cancel

NUMBER_OF_NEGATIVE_RATE	FLIGHT_NUMBER
6	AM123
1	YM445

4-List Flights that have best feedback?

```sql
SELECT COUNT(C.PASSENGER_SATISFACTION_RATE)  AS
NUMBER_OF_POSITIVE_FEEDBACKS ,F.FLIGHT_NUMBER
FROM
CUSTOMER_RATE_FACT  C
JOIN FLIGHT_DIMENSION F
ON
C.FLIGHT_KEY=F.FLIGHT_KEY
JOIN CUSTOMER_SERVICES_DIMENSION S
ON
S.CUSTOMER_SERV_KEY=C.CUSTOMER_SERV_KEY
WHERE C.PASSENGER_SATISFACTION_RATE >= 7 AND S.DESCRIPTION ='FEEDBACK'
GROUP BY F.FLIGHT_NUMBER
ORDER BY NUMBER_OF_POSITIVE_FEEDBACKS DESC;
```

```sql
--List Flights that have best feedback?
SELECT COUNT(C.PASSENGER_SATISFACTION_RATE)  AS NUMBER_OF_POSITIVE_FEEDBACKS ,F.FLIGHT_NUMBER
FROM
CUSTOMER_RATE_FACT  C
JOIN FLIGHT_DIMENSION F
ON
C.FLIGHT_KEY=F.FLIGHT_KEY
JOIN CUSTOMER_SERVICES_DIMENSION S
ON
S.CUSTOMER_SERV_KEY=C.CUSTOMER_SERV_KEY
WHERE C.PASSENGER_SATISFACTION_RATE >= 7 AND S.DESCRIPTION ='FEEDBACK'
GROUP BY F.FLIGHT_NUMBER
ORDER BY NUMBER_OF_POSITIVE_FEEDBACKS DESC;
```

Data Grid | Auto Trace | DBMS Output (disabled) | Query Viewer | CodeXpert | Explain Plan | Script Output

Cancel

NUMBER_OF_POSITIVE_FEEDBACKS	FLIGHT_NUMBER
3	AH435
3	OR569
2	NY123

5-List Top Loyal Frequent Flyers ?

```
SELECT COUNT(B.PASSENGER_KEY) AS NUMBER_OF_FLIGHTS,P.FIRST_NAME ,
P.LAST_NAME , P.PASSPORT_NUMBER ,F.CLASS_NAME
FROM PASSENGER_DIMENSION P
JOIN   FREQUENT_FLYER_CLASS_DIMENSION F
ON
P.FREQUENT_FLYER_CLASS_KEY=F.FREQUENT_FLYER_CLASS_KEY
JOIN  BOOKING_TRANSACTION_FACT B
ON
P.PASSENGER_KEY=B.PASSENGER_KEY
WHERE F.CLASS_NAME <> 'NON FF'
GROUP BY P.FIRST_NAME , P.LAST_NAME , P.PASSPORT_NUMBER ,F.CLASS_NAME
ORDER BY NUMBER_OF_FLIGHTS DESC;
```