

Data Warehouse Project



Airline Case Study

Mohamed Fathy

Salma Amr

Amr Elsayeh

Table of Contents

Introduction	3
Business process	3
Design process	4
Model.....	4
Granularity	4
Dimensions.....	4
Facts	4
Logical model design.....	5
Team Modeling Process	5
Business Question Answers.....	7
Creating Table Queries.....	10
Creating Dimensions Table.....	10
Creating Facts Tables	13
Decision Makers Questions?	17
Screenshots for Tables in Toad	21
Indexes and partitions	27
Indexes	27
B-Tree Indexes.....	28
Bitmap Join	28
Partitions.....	29
Horizontal partitioning	29
Vertical Partitioning.....	29
Functional Partitioning.....	30
References.....	31

Introduction

Business process

The case study is about airplane company which want to analysis it's flight activity We also need to track the marketing stage and support frequent flies in a special way to keep flying with the company.

The financial team want to track to analysis the company profit, and which airline is gaining more money.

Customer care want to track the customer opinion, complain, suggestions or inquiry in the good way

We need to install all the requirements into a data warehouse where we can answer all the business questions to develop the company's performance and gain more revenue, and of course raise the customer satisfaction level.

Design process

Model

We decided to work with Bill Inmon model to design the data warehouse.

Reason for the model:

The model of the airplane is new, and I think the business questions we need to cover doesn't cover all the business requirement, so for high percentage there will be more business requirements and reports needed for future development, so it will be much better to have a data stored in 3rd normal form, so we can easily develop a data mart to report from it. Inmon model will provide all the enterprise data, any future reports will be extracted from the data.

Granularity

1- F_Flight_Instance

The gain for this fact table is every flight instance of a single passenger.

2- F_Frequent_Flyer_Miles

The grain for this fact is every transaction or ride the frequent flyer made, either he redeems a reward or make more miles so his score increases.

3- F_Frequent_Flyer_Hotel

The grain for this fact is per booking in hotel, when a frequent flyer book night in a certain hotel for the number of nights during his transit.

4-F_Customer_Request

The grain is per customer request for every passenger.

5- F_Customer_Promotion

the grain for this fact is per redeem promotion for the frequent flyer.

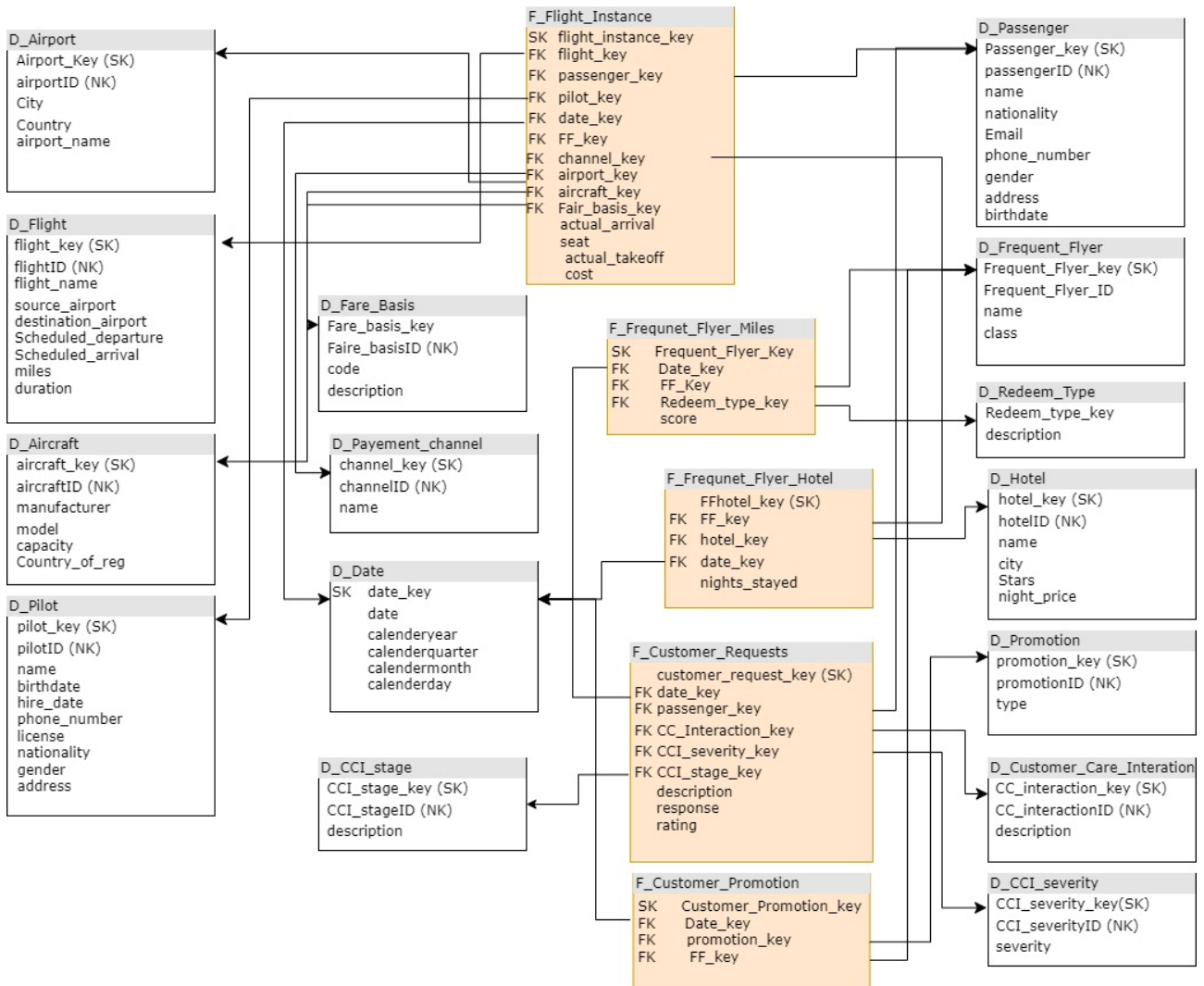
Dimensions		Facts
<ul style="list-style-type: none"> • D_Airport • D_Flight • D_Aircraft • D_Pilot • D_Payment_Channel • D_Date • D_CCI_Stag • D_Passenger 	<ul style="list-style-type: none"> • D_Frequent_Flyer • D_Redeem_Type • D_Hotel • D_Promotion • D_CC_Interaction • D_CCI_Severity 	<ul style="list-style-type: none"> • F_Flight_Instance • F_Frequent_Flyer_Miles • F_Frequent_Flyer_Hotel • F_Customer_Requests • F_Customer_Promotion

CC = customer Care

CCI = customer care interaction.

FF = Frequent Flyer

Logical model design



Team Modeling Process

We analyzed all the business requirements and installed it in our logical design and to do that we made some certain assumptions:

Assumptions:

- assumed** that every passenger has a passenger key and frequent flyer id in case he has a frequent flyer id, or null if he doesn't in the flight instance
- assumed** that marketing team manually or automatically with a certain code and bash script that every month check the miles traveled for every passenger and decide if his status is upgraded or degraded, also give him his score if he is a frequent flyer

- C. **assumed** if the passenger exceeds 5000 miles, he will gain the blue card and get a frequent flyer id, from 5000-30000 he will be silver, reaches 30000 with last 2 years is gold, reaches 300000 miles with the last 2 years is elite, reaches 600000 with the last 3 years is Platinum
- D. **assumed** that score is calculated depends on the frequent flyer class if he is
blue 1mile=1score.
if he is silver 1 mile = 2 score.
if he is gold 1 mile = 5 score.
if he is elite 1 mile = 10 score.
if he is platinum 1 mile = 20 score.
- E. **assumed** that with the score you can redeem an upgrade of the flight class from economic to class A for example.
Also, can redeem many rewards of shops related to the agency like clothes, devices etc. with certain score point.
Also, can redeem the score to book some days in certain hotels in the transit period.
- F. **assumed** that company has a contract with many hotels near the airport that transit occurs for the frequent flyers only, and its price is added to the ticket with a very small fare, also can be redeemed with frequent flyer score system, and we record the price of night that we added on ticket, class of hotel in case that frequent flyer want to increase the hotel class.
- G. **assume** that reservation process can be done by several channels like application, offline from window, website or by calling.

Business Question Answers

1- Executives decided that the first deliverable should focus on the flight activity in order to ensure good ongoing business process.

- to track the flight activity, we decided to make flight instance fact, which describe the instance of every flight for every passenger to track all the questions of business we imagined
- for example, if we need to check the performance of a place to go in the planned time, we will join the fact table with the D_Flight dimension to compare the planned take off with the actual take off with a certain aircraft_key to check the performance of that aircraft and judge the maintenance team for it.
- to calculate the miles for every passenger so we can calculate total miles he did so we can upgrade him to his classes (either blue - silver - gold - elite - platinum) so we can upgrade him, give him Frequent flyer id , and give him the score based on miles.
- track the pilot's performance and how many flights he did every period.
- track the activity on seat, and which seats are highly booked to raise its fare based on other seats and increase the profit
- track the reservation process, and which passenger booked either from online website, or application, or offline from the window
- track the activity of every airport, for example the company may have 2 or more airports in same city or country, and want to check which gain more profit, and how to increase the profit for other airports.
- calculate the profit for every flight and check if night flights gain more or morning flight by queries.
- Calculate the amount of foreign nationally travels to know if the high percentage of travelers and foreigners or people traveling for work so we can make marketing strategies based on that, for example if there is a lot of tourists, we can make a shop that sell historical products to attract them to travel with our agency and give us high rating

2- The marketing department wants to analyze what flights the company's frequent flyers take, what fare basis they pay, how often they upgrade, how they earn and redeem their frequent flyer miles, whether they respond to special fare promotions, how long their overnight stays are, and what proportion of these frequent flyers have gold, platinum, or titanium status.

a- what flights the company's frequent flyers take

we can easily know what flights the company frequent flyers take from instance of fact; with the frequent flyer id and the flight id we can track every frequent flyer he takes.

b- what fare basis they pay

in the instance flight fact, we can easily map the frequent flyer key with the fare basis he pays for every flight, to know how much he pay and which flight class he is usually engaged to, and either we should give him some promotion to encourage him to pay more money in the flight or not.

c- how often they upgrade

here we made a promotion fact table, so every time the frequent flyer upgrades his flight class with a promotion it will be mentioned there, so if we want how often they upgrade we need to map this fact table with instance of table using the frequent flyer dimension, so if the frequent flyer key was mentioned in both facts, we can count it to know how often they upgrade.

d- how they earn and redeem their frequent flyer miles

we made a score system as mentioned in assumption, so every frequent flyer make a new travel , we insert a row in the F_Frequent_Flyer_Miles with the type of the transaction (either its redeeming or a new flight's miles) to add or subtract his score, so every frequent flyer has a current score of him, and he can use it as assumption mentioned in several places, like promotion to upgrade flight ticket, or buy prices with the score (clothes and devices) from the company sponsors we can get even more money from our sponsors by giving them our customers!

so the frequent flyer miles track the score of every frequent flyer (note: you can't get score if you aren't a frequent flyer).

e- whether they respond to special fare promotions

as mentioned before we can map the promotion fact with an instance of fact with frequent flyer dimension to know if the redeems a promotion, also we can map frequent_flyer_miles with the instance of fact using the frequent flyer dimension to check if the type of redeem is negative then he responded to the special fare promotions, also can count how many times he responded easily (we will group with the frequent_flyer_mile)

f- how long their overnight stays are

as said in assumptions, we made a hotel fact table to record every time a frequent flyer stayed at the hotel and that hotel system is made for frequent flyers only to encourage people to be frequent flyers and travel a lot , for how many nights, so we can record the overnight stays of the frequent flyer for example during a period of time, like year 2020 the specific frequent flyer spent how many nights in hotel , and with class of

hotel or price we can know how frequent that flyer can pay, so if he prefer class A in hotels, we can give him marketing strategy to pay a lot more money since he is rich!.

g- what proportion of these frequent flyers have gold, platinum, or titanium status.

we can easily get this from dimension of the frequent flyer as it has the status of them, and the id of the frequent flyer

3- Your analysis shall also include the reservation process where the finance team will be interested in analyzing the company profit. Note that reservation processes can take place through multiple channels.

this can be calculated from the instance fact table as it has channel key mapped to the channel dimension so every passenger has a channel key that describe which channel he made to fly (channel is website, calling, offline window, application) so we can know which channel gives more profit, and try to market the other channels, also can reward the employees in that channel for doing good job and encourage them to work harder.

4- Airline company also provides customer care interaction before, within and after your trip in order to handle customer inquiries, complaints and keep their feedback for business enhancements.

Analysis should include interaction type and problem severity (if issue exists)

we made a F_Customer_request which tracks the customer requests, and it has several measures, connected to 3 dimensions

a- D_CC_Interaction which describe the interaction of the customer and has the type of it wither it's complained or inquiry or suggestion or opinion

b- D_CCI_Severity: which describes how urgent the interaction is, if it's urgent or normal so if it's urgent it should be answered faster and has higher priority.

c- D_CCI_Stage: which describe the stage of the interaction if it's before the flight as the customer has urgent inquiry or during the flight or after the flight

here we can easily track the customer request and know how usually important inquiry happens, or how bad the complains are, if the customer care should focus on complains during the flight or before or after? also there is columns for description which describe the complain of the customer, and it's response, and rating for the customer for its response which is very important to track the performance of the customer care and how they satisfied the customer.

Creating Table Queries

Creating Dimensions Table

```
Create table D_aircraft(  
AirCraft_key number(6),  
aircraft_ID number(6) not null,  
Manufacturer varchar2(200),  
Model varchar2(200) not null,  
Capacity number(3),  
country_of_register varchar2(100),  
CONSTRAINT pk_aircraft PRIMARY KEY (AirCraft_key)  
)
```

```
Create table D_Flight(  
Flight_Key number(6),  
Flight_ID number(6) not null,  
Flight_name varchar2(200) not null,  
Source_Airport varchar2(200) not null,  
Destination_Airport varchar2(200) not null,  
SCHEDULED_departure date not null,  
SCHEDULED_arrival date not null,  
Duration number(4),  
Miles number(4) not null,  
CONSTRAINT pk_flight PRIMARY KEY (Flight_Key)  
)
```

```
Create table D_pilot(  
pilot_key number(6),  
pilot_ID number(6) not null,  
name varchar2(250),  
hire_date date not null,  
phone_number varchar2(20) not null,  
license varchar2(30) not null,  
address varchar2(200),  
birth_date date,  
nationality varchar2(200),  
gender varchar2(50),  
CONSTRAINT pk_pilot PRIMARY KEY (pilot_key)  
)
```

```
Create table D_passenger(  
Passenger_key number(6),
```

```
Passenger_ID number(6) not null,  
name varchar2(250),  
phone_number varchar2(20) not null,  
address varchar2(200),  
birth_date date,  
nationality varchar2(200),  
gender varchar2(50),  
Email varchar2(100),  
CONSTRAINT pk_passenger PRIMARY KEY (Passenger_key)  
)
```

```
Create table D_frequent_flyer(  
Frequent_Flyer_key number(6),  
Frequent_Flyer_ID number(6) not null,  
name varchar2(250),  
class varchar2(20) not null,  
CONSTRAINT pk_Frequent_Flyer PRIMARY KEY (Frequent_Flyer_key)  
)
```

```
Create table D_promotion(  
promotion_key number(6),  
Promotion_ID number(6) not null,  
type varchar2(100),  
CONSTRAINT pk_promotion PRIMARY KEY (promotion_key)  
)
```

```
Create table D_customer_care(  
cc_interaction_key number(6),  
cc_ID number(6) not null,  
type varchar2(100),  
CONSTRAINT pk_cc PRIMARY KEY (CC_interaction_key)  
)
```

```
Create table D_CCI_Severity(  
CCI_Severity_Key number(6),  
CCI_Severity_ID number(6) not null,  
type varchar2(100),  
description varchar2(1000),  
CONSTRAINT pk_cci_severity PRIMARY KEY (CCI_Severity_Key)  
)
```

```
Create table D_CCI_Stage(  
CCI_Stage_Key number(6),  
CCI_Stage_ID number(6) not null,  
type varchar2(100),  
CONSTRAINT pk_cci_stage PRIMARY KEY (CCI_Stage_Key)  
)
```

```
Create table D_Hotel(  
Hotel_Key number(6),  
Name varchar2(250) not null,  
City varchar2(100),  
Stars number(7) not null,  
Price_Per_Night number(6) not null,  
CONSTRAINT pk_hotel PRIMARY KEY (Hotel_Key)  
)
```

```
Create table D_channel(  
Channel_Key number(6),  
Name varchar2(250) not null,  
CONSTRAINT pk_channel PRIMARY KEY (Channel_Key)  
)
```

```
Create table D_Airport(  
Airport_Key number(6),  
Airport_ID number(6) not null,  
Name varchar2(250) not null,  
City varchar2(100),  
Country varchar2(100),  
CONSTRAINT pk_Airport PRIMARY KEY (Airport_Key)  
)
```

```
Create table D_Redeem(  
Redeem_Type_Key number(6),  
description varchar2(100),  
CONSTRAINT pk_Redeem PRIMARY KEY (Redeem_Type_Key)  
)
```

```
Create table D_Fare_basis(  
Fare_basis_key number(6),  
Fare_basis_Id number(6) not null,  
code char(1) not null,  
description varchar2(200),  
CONSTRAINT pk_fare_basis PRIMARY KEY (Fare_basis_key)  
)
```

```
create table d_date(  
date_key number(6) primary key,  
date_date date
```

)

```

CREATE TABLE dateDimension AS
SELECT
n AS Date_key,
TO_DATE('31/12/2020','DD/MM/YYYY') + NUMTODSINTERVAL(n,'day') AS
Full_Date,
TO_CHAR(TO_DATE('31/12/2020','DD/MM/YYYY') +
NUMTODSINTERVAL(n,'day'),'DD') AS Days,
TO_CHAR(TO_DATE('31/12/2020','DD/MM/YYYY') +
NUMTODSINTERVAL(n,'day'),'Mon') AS Month_Short,
TO_CHAR(TO_DATE('31/12/2020','DD/MM/YYYY') +
NUMTODSINTERVAL(n,'day'),'MM') AS Month_Num,
TO_CHAR(TO_DATE('31/12/2020','DD/MM/YYYY') +
NUMTODSINTERVAL(n,'day'),'Month') AS Month_Long,
TO_CHAR(TO_DATE('31/12/2020','DD/MM/YYYY') +
NUMTODSINTERVAL(n,'day'),'YYYY') AS Year
FROM (
select level n
from dual
connect by level <= 9495)

```

Creating Facts Tables

```

Create table F_Flight_Instance(
Flight_Instance_key number(6),
Flight_key number(6),
passenger_key number(6),
source_airport_key number(6),
destination_airport_key number(6),
pilot_key number(6),
frequent_flyer_key number(6),
channel_key number(6),
aircraft_key number(6),
date_key number(6),
fare_basis_key number(6),
seat number(3),
cost number(6),
delay number(6),

```

```

constraint pk_flight_instance primary key (Flight_Instance_key),
constraint fk_flight foreign key (Flight_key) references D_flight(FLIIGHT_KEY),

```

```

constraint fk_passenger foreign key (passenger_key) references
D_passenger(PASSENGER_KEY),
constraint fk_s_airport foreign key (source_airport_key) references
D_airport(AIRPORT_KEY),
constraint fk_d_airport foreign key (destination_airport_key) references
D_airport(AIRPORT_KEY),
constraint fk_pilot foreign key (pilot_key) references D_pilot(PILOT_KEY),
constraint fk_frequent_flyer foreign key (frequent_flyer_key) references
D_frequent_flyer(FREQUENT_FLYER_KEY),
constraint fk_channel foreign key (channel_key) references
D_channel(CHANNEL_KEY),
constraint fk_aircraft foreign key (aircraft_key) references
D_aircraft(AIRCRAFT_KEY),
constraint fk_date foreign key (date_key) references D_date(DATE_KEY),
constraint fk_fare_basis foreign key (fare_basis_key) references
D_fare_basis(FARE_BASIS_KEY)
);

```

```

Create table F_Customer_Request(
Customer_request_key number(6),
date_key number(6),
passenger_key number(6),
CC_interaction_key number(6),
CCI_severity_key number(6),
CCI_Stage number(6),
description varchar2(1000),
response varchar2(1000),
rating number(1),

```

```

constraint pk_customer_request primary key (Customer_request_key),
constraint fk_date_cc foreign key (date_key) references D_date(DATE_KEY),
constraint fk_passenger_cc foreign key (passenger_key) references
D_passenger(PASSENGER_KEY),
constraint fk_cc_interaction_key foreign key (CC_interaction_key) references
D_customer_care(CC_INTERACTION_KEY),
constraint fk_CCI_severity foreign key (CCI_severity_key) references
D_cci_severity(CCI_SEVERITY_KEY),
constraint fk_CCI_Stage foreign key (CCI_Stage) references
D_cci_stage(CCI_STAGE_KEY)
);

```

```

Create table F_frequent_flyer_miles(
frequent_flyer_miles number(6),
date_key number(6),

```

```
frequent_flyer_key number(6),  
redeem_type number(6),  
rating number(6),
```

```
constraint pk_frequent_flyer_miles primary key (frequent_flyer_miles),  
constraint fk_date_ff_miles foreign key (date_key) references D_date(DATE_KEY),  
constraint fk_frequent_flyer_miles foreign key (frequent_flyer_miles) references  
D_frequent_flyer(FREQUENT_FLYER_KEY),  
constraint fk_redeem foreign key (redeem_type) references  
D_redeem(REDEEM_TYPE_KEY)  
);
```

```
Create table F_frequent_flyer_hotel(  
frequent_flyer_hotel number(6),  
date_key number(6),  
frequent_flyer_key number(6),  
hotel_key number(6),  
nights_stayed number(6),
```

```
constraint pk_frequent_flyer_hotel primary key (frequent_flyer_hotel),  
constraint fk_date_hotel foreign key (date_key) references D_date(DATE_KEY),  
constraint fk_frequent_flyer_hotel foreign key (frequent_flyer_key) references  
D_frequent_flyer(FREQUENT_FLYER_KEY),  
constraint fk_hotel foreign key (hotel_key) references D_hotel(HOTEL_KEY)  
);
```

```
create table F_Customer_promotion(  
Customer_promotion_key number(6),  
date_key number(6),  
frequent_flyer_key number(6),  
promotion_key number(6),
```

```
constraint pk_customer_promotion primary key (Customer_promotion_key),  
constraint fk_date_promotion foreign key (date_key) references  
D_date(DATE_KEY),  
constraint fk_frequent_flyer_promotion foreign key (frequent_flyer_key) references  
D_frequent_flyer(FREQUENT_FLYER_KEY),  
constraint fk_promotion foreign key (promotion_key) references  
D_promotion(PROMOTION_KEY)  
);
```

```
Create table F_customer_promotion(  
customer_promotion_key number(6),  
date_key number(6),  
frequent_flyer_key number(6),  
promotion_key number(6),
```

```
constraint pk_customer_promotion primary key (customer_promotion_key),  
constraint fk_date_promotion foreign key (date_key) references  
D_date(DATE_KEY),  
constraint fk_frequent_flyer_promotion foreign key (promotion_key) references  
D_promotion(PROMOTION_KEY)  
);
```

```
ALTER TABLE f_flight_instance  
ADD CONSTRAINT chk_seat CHECK (seat between 1 and 210);  
ALTER TABLE f_flight_instance  
ADD CONSTRAINT chk_delay CHECK (delay between 0 and 120);  
ALTER TABLE f_flight_instance  
ADD CONSTRAINT chk_cost CHECK (cost between 100 and 1000);
```

Decision Makers Questions?

The business wants to know the highest 3 airport that made profit for the company during january 2022 assuming that the profit is a fixed amount of 30%.

```
select * from (select sum(cost)*0.3 as Profit , air.name as Airport_name
from f_flight_instance f join d_date using(DATE_KEY)
join d_airport air on F.SOURCE_AIRPORT_KEY = AIR.AIRPORT_KEY
where month_num = 01 and year =2022
group by air.name
order by 1 desc) where rownum < 4;
```

PROFIT	AIRPORT_NAME
2268	Almaza Airport
1989	Sphinx International Airport
1652.4	Hurghada Airport

The HR decided to award the best performed pilot in 2022 depends on the number of flight since he was available for all highest number of flights, and assigned the DWH team to get his name from the data warehouse.

```
select * from (select p.name, count(f.PILOT_KEY) "number of Flights"
from d_pilot p join f_flight_instance f on P.PILOT_KEY = F.PILOT_KEY
join d_date d on D.DATE_KEY = F.DATE_KEY
where year = 2022
group by p.name
order by 2 desc)
where rownum = 1;
```

Output	Grid 1	Environment
	NAME	number of Flights
	Fatma	14

The Company wants to know the flights that delayed more than an hour in the last December assigned with its date because there were many complains about the delay of the flights.

```
select fl.flight_name as Flight_name, d.full_date as dat
from d_flight fl join f_flight_instance f on FL.FLIIGHT_KEY = F.FLIGHT_KEY join d_date d on
D.DATE_KEY = F.DATE_KEY
where delay > 100 and month_num = 12 ;
```

FLIGHT_NAME	DAT
Silkroad	04-DEC-21
Silkroad	24-DEC-21
Silkroad	31-DEC-20
Matrouh	23-DEC-21
UpperEgy	28-DEC-21
UpperEgy	17-DEC-21
Matrouh	09-DEC-21
UpperEgy	24-DEC-21
Silkroad	25-DEC-21
Matrouh	15-DEC-21
UpperEgy	20-DEC-21
Matrouh	10-DEC-21
Silkroad	06-DEC-21

The Company wanted to know for every airport in the company, which was the highest channel used by passengers in October 2021.

```
select a.name "Airport Name",c.name "Channel", count(*) "Number of Passengers"
from d_channel c join f_flight_instance f on C.CHANNEL_KEY = F.CHANNEL_KEY join d_airport a on
A.AIRPORT_KEY=F.SOURCE_AIRPORT_KEY join d_date d on d.date_key = F.DATE_KEY
where month_num = 10 and year = 2021
group by a.name, c.name
order by 3 desc, 1,2 ;
```

Airport Name	Channel	Number of Passengers
Alexandria International Airport	Offline	6
Cairo Airport	Website	6
Abu Simbel Airport	Mobile Application	5
Almaza Airport	Mobile Application	5
Almaza Airport	Website	4
Assiut Airport	Mobile Application	4
El Gouna Airport	Mobile Application	4
Abu Simbel Airport	Call	3
Almaza Airport	Call	3
Almaza Airport	Offline	3
Aswan Airport	Offline	3
Cairo Airport	Mobile Application	3
Cairo Airport	Offline	3
Sphinx International Airport	Offline	3
Abou Redis Airport	Call	2
Abou Redis Airport	Offline	2
Abu Simbel Airport	Website	2
Alexandria International Airport	Website	2
Assiut Airport	Offline	2

The Marketing department wanted to award the frequent flyer who traveled the highest distance during 2022 to raise his satisfaction level and loyalty towards the company.

```
select ff.name , sum(FL.MILES) "Number of Miles"
from d_frequent_flyer ff join f_flight_instance f on FF.FREQUENT_FLYER_KEY =
F.FREQUENT_FLYER_KEY
join d_date d on d.date_key = F.DATE_KEY join d_flight fl on FL.FLIGHT_KEY = F.FLIGHT_KEY
where year = 2021
group by ff.name
order by 2 desc;
```

Output	Grid 1	Environment
NAME	Number of Miles	
Salma Amr	137600	
Mohamed Fathy	129000	
Amr Elsayeh	124500	
Yomna	112400	

The Marketing team wanted to make sure that all frequent flyers used the promotion keys given by the marketing department, and his names.

```
select FF.NAME, FF.FREQUENT_FLYER_ID
from d_frequent_flyer ff left outer join f_customer_promotion p on FF.FREQUENT_FLYER_KEY =
P.FREQUENT_FLYER_KEY
where PROMOTION_KEY is null ;
```

Output	Grid 1	Environment
NAME	FREQUENT_FLYER_ID	
Yousef	5	

Due to high financial fees of hotels, the company wanted to disclude the hotel where the frequent flyers don't frequent use, so we want to know the name of hotels with overnight stayed ascending.

```
select H.NAME, H.CITY,FF.NIGHTS_STAYED
from f_frequent_flyer_hotel ff join d_hotel h on FF.FREQUENT_FLYER_HOTEL = H.HOTEL_KEY
group by H.NAME,H.CITY,FF.NIGHTS_STAYED
order by 3
```

NAME	CITY	NIGHTS_STAYED
FourSeasons	Alex	0
Sun Rise	Hurghada	1
Dusit	Cairo	3
Hilton	Cairo	3

The marketing department wanted to know the activity of frequent fliers and how many times every frequent flyer used our offers to redeem rewards so we can increase offers for people who frequently redeem rewards.

```
select FF.NAME, count(FFM.REDEEM_TYPE) "Number of Redeems"
from d_frequent_flyer ff join f_frequent_flyer_miles ffm on FF.FREQUENT_FLYER_KEY =
FFM.FREQUENT_FLYER_KEY
where FFM.REDEEM_TYPE = 2
group by ff.name
order by 2 desc
```

NAME	Number of Redeems
Amr Elsayeh	140
Salma Amr	138
Yomna	130
Mohamed Fathy	102

The Customer server manager was so mad because too many bad rating last December for the response he makes, and wanted to know the interaction with rating from 1 to 3 to revise it, he wanted to know the passenger name and his issue and it's response.

```
select p.name, FCR.DESCTIPTION, FCR.RESPONSE ,FCR.RATING
from f_customer_request fcr join d_passenger p on FCR.PASSENGER_KEY = P.PASSENGER_KEY join
d_date d on D.DATE_KEY = FCR.DATE_KEY
where month_num = 12 and FCR.RATING between 1 and 3;
```

NAME	DESCRIPTIION	RESPONSE	RATING
Amr Khairy	KikcqtqBhtHSONvhMXUIPhJVfKqAqEftYvONE sxPZIUWqxpZSJNlfGxzaztkdGUBUvzodHsZJ LPXdjrDqxVpsZVsnZsBtCSAaRegxEsnAngy nwnpkjMmIPYAgamHDrMoFDgposNAUifpUG BdEBzBblgOSdgyQuHgwWYhZFFpCtpxbxBdy XjQSSyYhayHckMloaUfNyAeHeBGWGJWRH LTxUHLfUcwWIngPETOwGjkPNageUCIFDPGn aUDsdykkWAPIpRUDnTVdhnWiyJHFxxGxsb XXiiYWSTExAxxrcWrajfGcHqKLO	UJwktbTwSASILrngakqzDkMyPKIZJSQnhidA ERJUwvYRYIFvXyApj	1
Ahmed Mostafa	hLKBbiQcMxLIHYIjWcpDjFPpFKAXUcaThO LTqmqjCFDWcKqIWyMzdGnzKzVzAUhjCVJ HrqveLGMxLNZbfgTEwyrRJBjXNvzSLnqkyJj hpMSUyXgcKeZotaKwEWIuKtUERrTdiVDxCJ uiyDdHwLMHigcRuDzahgPUFSTueIGyJWJm GwQdwhGwrKLXCbjfDyknhPkZeDvbUNUnB TWzpMuZEGedzLdiiNKbayfplogwpiZBxNBVe hkqPRFVCQnOQSyunuftLzZoZWpXuwrpq	eg	3
Ahmed Samir	TgnPazkqkVdNtUrjnLMcJxwLXlhKXtNASyBm GzSpqFmMdtZppkyZptyMDIEMhVmmKqxXB AnxtbzgnqoHxmCOdnqwxrEQSIDwoVkfwrzc kncYdujAaAhytyYqQIouCdObmOhhThOMKJ HzjystvGUXJgOMKfRXeprpgmDQzBdSUIYHE cegkddWJHuRizaVcJqcNbsahBBUzmnJWpC rXbXzrPRObrRIGUBwfrRaJeiGIWWZCwXNhZ	YTkULBoibXzhDHGhqLftWHLVvaXEeH	2

Screenshots for Tables in Toad

Aircraft dimension table

AIRCRAFT_KEY	AIRCRAFT_ID	MANUFACTURER	MODEL	CAPACITY	COUNTRY_OF_REGISTER
1	7	Boeing	737-886	215	Egypt
2	8	Boeing	737-800	215	Egypt
3	9	Boeing	737-700	215	Egypt
6	10	Boeing	737-400	200	Egypt
7	11	Airbus	A330	201	Egypt
8	12	Airbus	A352	260	Egypt
9	13	Airbus	A380	195	Egypt
4	18	Boeing	737-600	215	Egypt
5	19	Boeing	737-500	215	Egypt

Airport dimension table

AIRPORT_KEY	AIRPORT_ID	NAME	CITY	COUNTRY
1	12	Cairo Airport	Cairo	Egypt
2	13	Sphinx International Airport	6th of October	Egypt
3	14	Abou Redis Airport	Abou Redis	Egypt
4	15	Abu Simbel Airport	Abu Simbel	Egypt
5	16	Alexandria International Airport	Alexandria	Egypt
6	17	Almaza Airport	Cairo	Egypt
7	18	Assiut Airport	Assiut	Egypt
8	19	Aswan Airport	Aswanr	Egypt
9	20	El Gouna Airport	El Gouna	Egypt
10	21	Hurghada Airport	Hurghada	Egypt

CCI Severity dimension table

CCI_SEVERITY_KEY	CCI_SEVERITY_ID	TYPE	DESCRIPTION
1	1	P1	A critical incident with very high impa
2	2	P2	A major incident with significant impact
3	3	P3	A minor incident with low impact

CCI Stage dimension table

CCI_STAGE_KEY	CCI_STAGE_ID	TYPE
1	1	Before
2	3	During
3	4	After

Channel dimension table

CHANNEL_KEY	NAME
1	Offline
2	Website
3	Mobile Application
4	Call

CC interaction dimension table

CC_INTERACTION_KEY	CC_ID	TYPE
1	1	Question
2	2	Suggestion
3	3	Complaint

Date dimension table

DATE_KEY	FULL_DATE	DAYS	MONTH_SHORT	MONTH_NUM	MONTH_LONG	YEAR	HOURS
37271	01/01/2022 02:00:00	01	Jan	01	January	2022	2
37272	01/01/2022 03:00:00	01	Jan	01	January	2022	3
37273	01/01/2022 04:00:00	01	Jan	01	January	2022	4
37274	01/01/2022 05:00:00	01	Jan	01	January	2022	5
37275	01/01/2022 06:00:00	01	Jan	01	January	2022	6
37276	01/01/2022 07:00:00	01	Jan	01	January	2022	7
37277	01/01/2022 08:00:00	01	Jan	01	January	2022	8
37278	01/01/2022 09:00:00	01	Jan	01	January	2022	9
37279	01/01/2022 10:00:00	01	Jan	01	January	2022	10
37280	01/01/2022 11:00:00	01	Jan	01	January	2022	11
37281	01/01/2022 12:00:00	01	Jan	01	January	2022	12
37282	01/01/2022 13:00:00	01	Jan	01	January	2022	13
37283	01/01/2022 14:00:00	01	Jan	01	January	2022	14
37284	01/01/2022 15:00:00	01	Jan	01	January	2022	15
37285	01/01/2022 16:00:00	01	Jan	01	January	2022	16
37286	01/01/2022 17:00:00	01	Jan	01	January	2022	17
37287	01/01/2022 18:00:00	01	Jan	01	January	2022	18
37288	01/01/2022 19:00:00	01	Jan	01	January	2022	19
37289	01/01/2022 20:00:00	01	Jan	01	January	2022	20
37290	01/01/2022 21:00:00	01	Jan	01	January	2022	21
37291	01/01/2022 22:00:00	01	Jan	01	January	2022	22
37292	01/01/2022 23:00:00	01	Jan	01	January	2022	23
37293	02/01/2022	02	Jan	01	January	2022	0

Fare basis dimension table

FARE_BASIS_KEY	FARE_BASIS_ID	CODE	DESCRIPTION
1	1	F	First Class
2	2	B	Business Class
3	3	E	Economy Class

Flight Dimension table

FLIGHT_KEY	FLIGHT_ID	FLIGHT_NAME	SOURCE_AIRPORT	DESTINATION_AIRPORT	SCHEDULED_DEPARTURE	SCHEDULED_ARRIVAL	DURATION	MILES
1	1	Silkroad	Cairo Airport	El Gouna	29/12/2022 00:45:04	31/12/2022 00:45:04	2	500
2	3	Silkroad	Cairo Airport	El Gouna	29/12/2022 00:51:52	29/12/2022 02:51:52	2	500
3	4	UpperEgy	Cairo Airport	Aswan	29/12/2022 00:52:29	29/12/2022 03:52:29	2	600
4	6	Matrouh	Cairo Airport	Matrouh	29/12/2022 00:53:04	29/12/2022 03:53:04	2	900
5	7	SummerTime	Cairo Airport	Alex Airport	29/12/2022 00:53:32	29/12/2022 01:53:32	2	200
6	22	S4 leage	El gouna	Cairo Airport	29/12/2022	30/12/2022	1	600

Frequent Flyer Dimension table

FREQUENT_FLYER_KEY	FREQUENT_FLYER_ID	NAME	CLASS
1	1	Amr Elsayeh	Gold
2	2	Salma Amr	Gold
3	3	Mohamed Fathy	Gold
4	4	Yomna	Gold
5	5	Yousef	Gold

Hotel dimension table

HOTEL_KEY	NAME	CITY	STARS	PRICE_PER_NIGHT
1	FourSeasons	Alex	7	1000
2	Hilton	Cairo	5	800
3	Sun Rise	Hurghada	5	300
4	Dusit	Cairo	5	700

Passenger dimension table

PASSENGER_KEY	PASSENGER_ID	NAME	PHONE_NUMBER	ADDRESS	BIRTH_DATE	NATIONALITY	GENDER	EMAIL
4	13	Ahmed Samir	1111111	Nasr City		Egyptian	Male	ahmed@hotmail.com
5	14	Ahmed Mostafa	1111111	Nasr City		Egyptian	Male	mostafa@hotmail.com
6	15	Amr Khairy	1111111	Nasr City		Egyptian	Male	amr@hotmail.com
7	16	Marwan yasser	111111	Nasr City		Egyptian	Male	marwan@hotmail.com
8	17	Toaa Dawoud	111111	Nasr City		Egyptian	Male	toaa@hotmail.com
1	18	Amr Elsayeh	1155433996	Nasr City		Egyptian	Male	Amrelsayeh@hotmail.com
2	26	Salma Amr	1155476810	Alex		Egyptian	Female	Salmaamr@hotmail.com
3	28	Mohamed Fathy	1156633133	New Cairo		Egyptian	Male	mohamedfathy@hotmail.com

Pilot dimension table

PILOT_KEY	PILOT_ID	NAME	HIRE_DATE	PHONE_NUMBER	LICENSE	ADDRESS	BIRTH_DATE	NATIONALITY	GENDER
1	1	Amr Elsayeh	08/09/2013	1155433996	LX3UZ2	Nasr City		Egyptian	Male
2	2	Amr1	08/09/2013	11111116	LX3UZ2	Nasr City		Egyptian	Male
3	3	Amr2	08/09/2013	1155555596	LX3UZ2	Nasr City		Egyptian	Male
4	4	Amr3	08/09/2013	1155433996	LX3UZ2	Nasr City		Egyptian	Male
5	5	ahmed	08/09/2013	1155433996	LX3UZ2	Nasr City		Egyptian	Male
6	6	Amr Kahiry	08/09/2013	1155433996	LX3UZ2	Nasr City		Egyptian	Male
7	7	Fatma	08/09/2013	1155433996	LX3UZ2	New Cairo		Egyptian	Female
8	21	Mohamed Fathy	01/01/2020	011111111	Lxaaaa	New Cairo		Egyptian	Male
9	22	Salma	01/01/2020	0123	LX	Alex		Egyptian	Female

Promotion dimension table

PROMOTION_KEY	PROMOTION_ID	TYPE	DESCRIPTION
1	1	Super	From Economy To Business
2	2	Elite	From Business To First
3	3	Unique	From Economy To First

Redeem dimension table

REDEEM_TYPE_KEY	DESCRIPTION
1	Travel
2	Purchase

Customer promotion fact table

CUSTOMER_PROMOTION_KEY	DATE_KEY	FREQUENT_FLYER_KEY	PROMOTION_KEY
893529	37493	1	3
722915	31418	4	2
74272	30941	1	3
820846	33460	4	2
174891	31644	2	1
843721	36697	4	1
411816	37648	2	2
904865	37317	2	2
184109	34960	3	1
128824	34778	1	3
304123	28660	4	2
292655	36041	3	2
904907	36012	2	2
386179	36994	1	1
321253	35155	1	1
513030	30365	2	2
491238	28616	3	3
804422	37597	3	3
675169	30143	4	3
267473	29941	4	3

Customer request fact table

CUSTOMER_REQUEST_KEY	DATE_KEY	PASSENGER_KEY	CC_INTERACTION_KEY	CCI_SEVERITY_KEY	CCI_STAGE	DESCRIPTION	RESPONSE	RATING
5001	29047	1	3	2	2	LenHoxPhaLMYQmIqgPBMIoCwWYYYfO...	nKuytaCsqjUxjeEagpaPkLUDS...	5
5006	30416	8	1	3	3	AEFGKoVkkUgzyVgaqPOTfVOWJkfNshy...	ebBEWsjFzirIBYkMlutrCxvQ...	2
5011	35957	2	3	1	2	unB8DfKhNCeLJuCcyAvIXdoiQaXiPynUr...	oddxWAGUYqrHSNoSujtweKY...	3
5016	37119	3	1	2	1	NTXZTYpiuWKRzXqggjGZTijzAvaXoLNUk...	THNHIOIesrF8wDAXIAArHrfVw...	3
5021	37955	5	1	1	1	TkuaZRITZNwxdTgnUInLWNdPDnruAGMq...	xDSBbkCHEJJmcVhCBcqJaottJ...	9
5026	32337	1	3	3	1	mldIcfYHkqydnzpJKLZOKTKeqEzDqxdW...	qWwMdVHozJrtWhcJBzooHOW...	3
5031	29970	5	2	2	2	FmFNSFXuZPzaWOCyNszzTjIQmXZQSB...	mOZVRDonFdqzHeJbJdjzHda...	5
5036	36185	5	1	3	1	CpRrUVfCaPeFCeVZuMTtSloInPpRpZOT...	vHMiCSBxEVqJDEECpezAsrJs...	8
5041	30990	7	3	3	1	zypVfjXwQSFkUcwAkpAeVEMzuCNIPKN...	ZlcaTnNsZqVFZkokUUKOXBo...	9
5046	34995	7	1	3	1	pxcdTyPmHJprXUERQPCBJZAXZp	JFOLWGjKJnxQzKBrmQpEVpvy...	6
5051	36793	3	2	3	3	yylnFXfyXanSEfEIZvCtwoCJMKMPQRbK...	JHXCPYSukutzKmOnsziXhM...	3
5056	37549	4	1	2	1	VbakTeziYRObVXRQBhJBuwszGdzlBAm...	JPBjQfbSbFvsJziAISFGcwOlstP...	0
5061	33089	6	3	2	1	pbxnSJwopGntoIhOIFsqZEEKZhzqRbcYbj...	HLjhlrVfnMWsLbAskSTTLHyvn...	6
5066	36051	1	3	1	3	jowfktOrLSRnxHVVONBaoyqBPLJicCndI...	XMbZyexofzQSNRgGbaNstjKBE...	2
5071	31225	8	3	1	2	oNLkttDOOfbOCsTmgLJtVdnvjWryPbeI...	HucnzNgFCPUTJedyOEVPqeR...	1
5076	29155	7	2	2	2	zPhDJDYXnOWqHzVATGuLnitQidcoDyuCI...	PlsCUwFSTYlqaxMXANXCcSvs...	4
5081	37002	1	2	2	3	fmodqErtvFjhWNNYwIBALYvkBAOsOEdc...	lKJmIUwycOeXwnfjtELERTAT...	7
5086	32882	1	1	2	2	dMrOPCNTndMGBuPWnSxHxbCBMfqp...	iTPvrPCguupGdLKnjAQlrQveP...	4
5091	28715	6	3	1	1	sRHODxQMIGAEMNGQHFCWtlbClUejt...	sxEONJChFmhoaMWhgSlpK...	2
5096	32132	5	3	1	3	lajdUTlvcXjOsgGHvfyi	teLjnRxBewqmyuPpQbpxWmE...	9
5101	30417	4	2	3	3	pdCWKWztwTqMrRCqMRxPmKslyogXeL...	nwpBgYGdChqjTHPoOCeDozx...	0
5106	31850	1	1	3	3	sjfxPeqOUnREOrDGTaDbjaqycCpZBBWq...	YwRPGsIMMMmMJPJFBSjchHnZk...	7
5111	30918	6	2	2	1	UcaINlrjdxiILfxqjGQNGXgreoPlvUMvcb...	okwEKnamtiDpIDzuTYImIvZN...	7

Flight instance fact table

FLIGHT_INSTANCE_KEY	FLIGHT_KEY	PASSENGER_KEY	SOURCE_AIRPORT_KEY	DESTINATION...	PILOT_KEY	FREQU...	CHANNEL_KEY	AIRCRAFT_KEY	DATE_KEY	FARE_BASIS_KEY	SEAT	COST	DELAY
1701	4	4	6	1	7	2	3	1	30484	3	201	369	18
1702	3	7	10	1	3	3	2	9	37908	3	112	786	72
1703	2	1	1	5	7	3	2	8	35352	1	93	272	67
1704	4	3	6	6	4	3	4	4	36204	1	124	650	72
1705	5	6	3	1	1	3	2	5	35265	1	138	410	100
1706	1	1	2	4	1	1	2	4	34535	2	161	959	64
1707	4	3	2	5	5	4	4	9	37690	1	98	103	3
1708	2	4	3	9	3	4	2	9	31704	2	13	653	107
1709	3	4	2	6	7	4	3	6	33418	2	40	875	57
1710	5	8	4	6	2	4	1	3	28985	1	164	959	91
1711	2	4	1	1	7	4	3	9	32025	2	60	325	53
1712	3	1	1	5	4	1	1	4	35977	2	47	163	59
1713	2	1	2	3	1	4	4	4	29257	2	12	395	98
1714	1	1	8	10	7	3	4	4	31745	3	76	979	52
1715	4	6	2	6	4	2	2	1	33168	3	67	337	98
1716	5	5	10	2	6	1	4	3	37321	2	110	816	72
1717	1	2	8	3	6	3	3	2	29170	2	21	484	78
1718	2	2	5	7	4	1	3	1	28553	2	23	949	41
1719	2	8	9	10	7	1	3	7	32131	2	66	618	28
1720	3	8	3	3	6	2	1	5	33419	1	49	383	4
1721	2	2	3	6	6	1	1	3	37430	2	100	138	102
1722	1	8	4	3	6	4	2	1	32038	3	96	529	96
1723	2	8	2	9	4	2	4	6	32043	3	76	931	104

Frequent flyer hotel fact table

FREQUENT_FLYER_HOTEL	DATE_KEY	FREQUENT_FLYER_KEY	HOTEL_KEY	NIGHTS_STAYED
1	29047	1	1	0
2	30425	1	1	3
3	35975	4	3	1
4	37146	2	4	3
5	34064	1	2	3
6	28488	1	1	2
7	30024	2	4	1
8	36248	2	1	2
9	31062	3	2	2
10	35076	1	3	1
11	36883	3	4	3
12	37648	3	4	3
13	33197	3	1	2
14	36168	2	3	0
15	31351	1	4	3
16	29290	4	2	1
17	37146	1	1	1
18	33035	2	4	2
19	28877	4	2	1
20	32303	4	2	1
21	30597	3	2	1
22	32039	2	1	1
23	31116	3	4	1

Frequent flyer miles fact table

FREQUENT_FLYER_MILES	DATE_KEY	FREQUENT_FLYER_KEY	REDEEM_TYPE	SCORE
10010	29047	1	1	12979
10020	30427	1	2	3846
10030	35979	1	2	16489
10040	37152	3	1	17412
10050	34072	2	1	523
10060	28498	2	2	16658
10070	30036	1	2	4853
10080	36262	1	1	10037
10090	31078	4	1	16839
10100	35094	2	1	14196
10110	36903	4	2	10452
10120	37670	1	2	3627
10130	33221	2	1	16552
10140	36194	4	1	11560
10150	31379	1	2	4406
10160	29320	4	1	2119
10170	37178	2	1	14465
10180	33069	1	2	13713
10190	28913	2	2	2386
10200	32341	3	2	13689
10210	30637	1	2	14305
10220	32081	4	1	4231
10230	31160	1	1	13363

Indexes and partitions

Indexes

Indexes in data warehouse is important to reduce the amount of time it takes for query, it's used when:

- The columns are often used as a condition in the query
- The column contains a wide range of value or a large number of null values
- Most queries expected to retrieve less than 5% of the rows
- Automatically created when define PRIMARY KEY or UNIQUE constraint

Types of indexes in data warehouses:

1) Bitmap

Bitmap indexes are primarily designed for data warehousing or environments in which queries reference large amount of data and have ad hoc queries, but rarely have a DML transaction, it can be used for several situations:

- The number of unique values is small compared to the number of table row, so there are repetitive values which will make queries take much long time.
- The indexed table is read only or not subject to a lot of modification DML statements
- Highly used in data warehouses because it consumes less storage compared to others as they are typically only a fraction of the size of the indexed data in the table.
- Very efficient maintenance during parallel DML and loads

In a regular index, this is achieved by storing a list of rowids for each key corresponding to the rows with that key value. In a bitmap index, a bitmap for each key value is used instead of a list of rowids.

Each bit in the bitmap corresponds to a possible rowid, and if the bit is set, it means that the row with the corresponding rowid contains the key value.

A mapping function converts the bit position to an actual rowid, so the bitmap index provides the same functionality as a regular index even though it uses a different representation internally.

If the number of different key values is small, bitmap indexes are very space efficient.

The advantages of using bitmap indexes are greatest for low cardinality columns

Example: gender column which has only male or female

CUSTOMER #	MARITAL_ STATUS	REGION	GENDER	INCOME_ LEVEL
101	single	east	male	bracket_1
102	married	central	female	bracket_4
103	married	west	female	bracket_2
104	divorced	west	male	bracket_4
105	single	central	female	bracket_2
106	married	central	female	bracket_3

All columns except the customer id column has low cardinality, so bitmap index will be suitable, while for the customer id column a b-tree index would give the best performance.

Let's take the region column as example:

REGION='east'	REGION='central'	REGION='west'
1	0	0
0	1	0
0	0	1
0	0	1
0	1	0
0	1	0

- Each entry (or bit) in the bitmap corresponds to a single row of the CUSTOMER table.
- The value of each bit depends upon the values of the corresponding row in the table.

If we want to know for example the number of our married customers who lives in the central or west regions, we will write the following query:

```
SELECT COUNT(*) FROM customer
```

```
WHERE MARITAL_STATUS = 'married' AND REGION IN ('central','west');
```

Bitmap indexes can process this query with great efficiency by merely counting the number of ones in the resulting bitmap.

B-Tree Indexes

It is designed as an upside-down tree, a tree that has hierarchy of root --> branch --> leaf where leaf is the entry point with the actual data (values and pointers to row). we use B-tree indexes when we know that our typical query refers to the indexed column and retrieves a few rows.

The advantage of B-tree indexes is that it is simple and suitable of exact match query, so it's used to indexes unique or semi unique columns.

B-tree indexes are most used in a data warehouse to index keys which are unique or near-unique.

In many cases, it may not be necessary to index these columns in a data warehouse, because unique constraints can be maintained without an index, and since typical data warehouse queries may not yield better performance with such indexes.

Bitmap indexes should be more common than B-tree indexes in most data warehouse environments.

Bitmap Join

Oracle has introduced this method to create speed join queries against very large data warehouse tables. it creates an index that performs the join at index creation time and creates a bitmap index of the keys that are used in the join.

Scenario	Query time (ms)	Cost (Explanation plan)	Index storage space (MBs)	Index build time (seconds)
0 - No indexes	2400	8248	0	0
1 - B-tree	<16	65	208 (fact table), 0.3 (date primary key)	26 (fact table), 0.06 (date primary key)
2 - Bitmap	<16	55	14 (1 dimension key in fact table), 0.06 (account attribute)	8 (1 dimension key in fact table), 0.03 (account attribute)
3 - Bitmap Join	<16	9	2 (for each attribute)	25 (for each attribute),

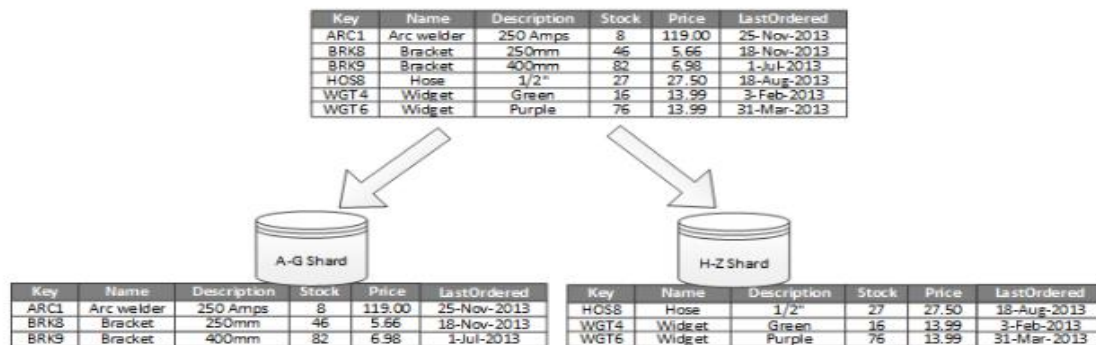
Partitions

Partitioning helps to scale a data warehouse by dividing database objects into smaller pieces, enabling access to smaller, easy management, assist backup/recover, enhances performance.

Types of partitioning:

Horizontal partitioning

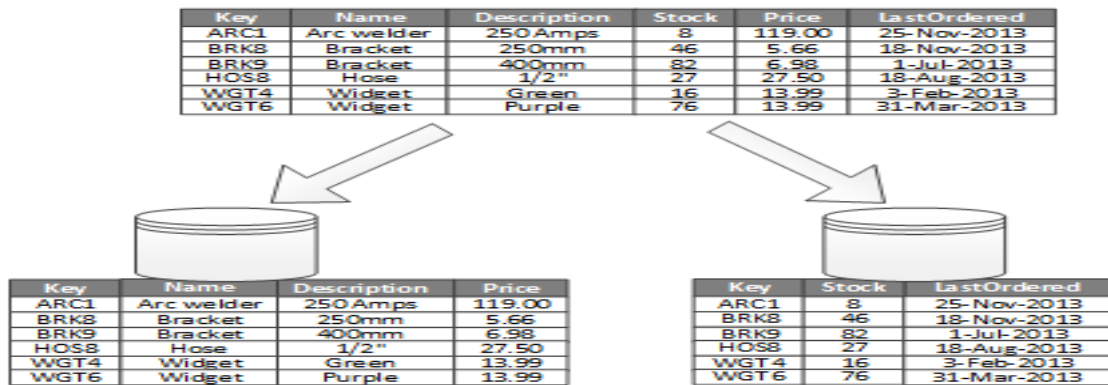
In this strategy, each partition is a separate data store, but all partitions have the same schema. Each partition is known as a shard and holds a specific subset of the data, such as



Vertical Partitioning

In this strategy, each partition holds a subset of the fields for items in the data store. The fields are divided according to their pattern of use. For example, frequently accessed fields might be placed in one vertical partition and less frequently accessed fields in another.

The most common use for vertical partitioning is to reduce the I/O and performance costs associated with fetching items that are frequently accessed.



Functional Partitioning

In this strategy, data is aggregated according to how it is used by each bounded context in the system. For example, an e-commerce system might store invoice data in one partition and product inventory data in another.

When it's possible to identify a bounded context for each distinct business area in an application, functional partitioning is a way to improve isolation and data access performance. Another common use for functional partitioning is to separate read-write data from read-only data.

References

<https://app.diagrams.net/#G1rof8AUctnr7WfyC9TFvsiRSKM2HymmNV>

<https://www.atlassian.com/incident-management/kpis/severity-levels>

<https://thepointsguy.com/guide/american-airlines-fare-classes/>

<https://www.jetnet.com/features/flight-activity/>

<https://www.oracle.com/database/technologies/airlines-data-model.html>

<https://www.slideshare.net/trivadis/data-warehousing-and-bitmap-indexes-more-than-just-some-bits-39738880>

<https://docs.oracle.com/database/121/DWHSG/schemas.htm#DWHSG9050>

<https://www.slideshare.net/trivadis/partitioning-your-oracle-data-warehouse-just-a-simple-task>

[Data partitioning guidance - Azure Architecture Center | Microsoft Learn](#)

[Data Warehousing - Partitioning Strategy \(tutorialspoint.com\)](#)