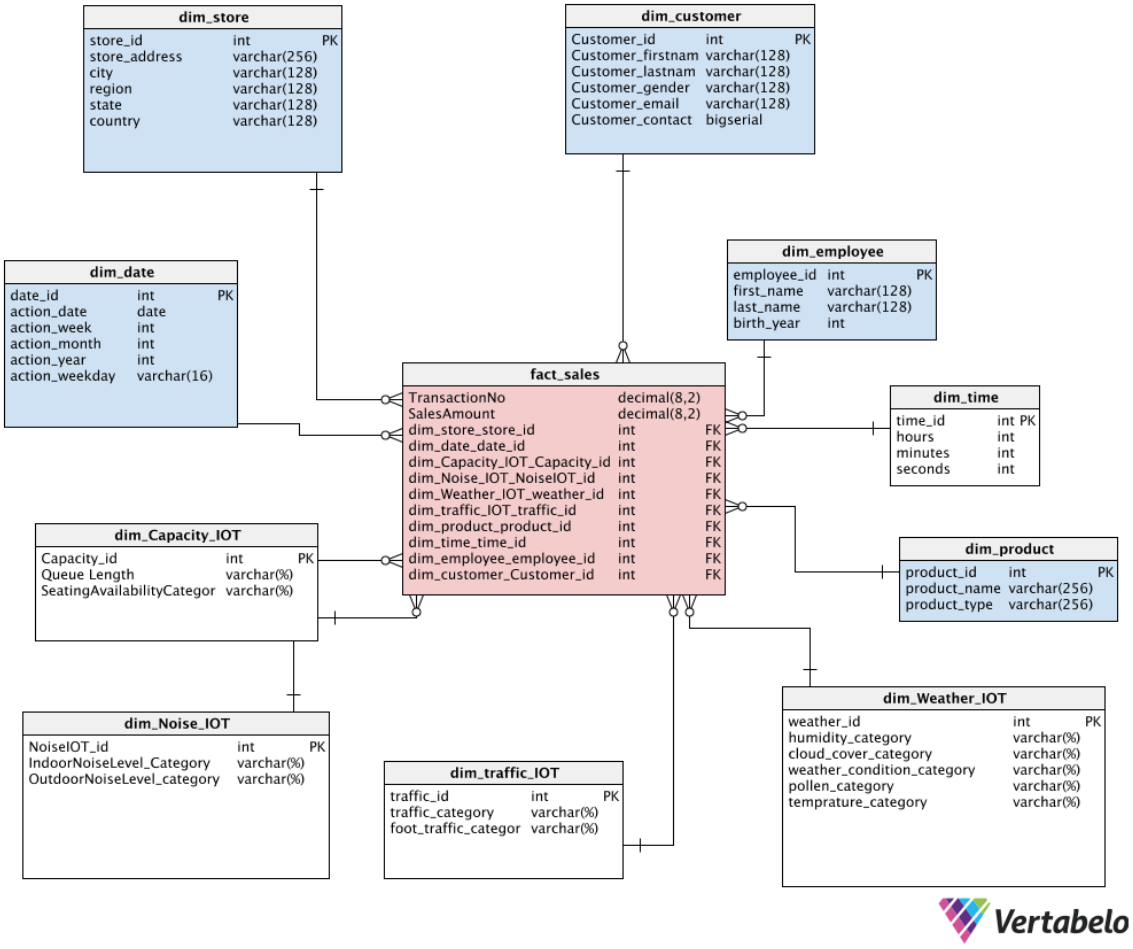
There are two granularities in this scenario. The original fact table has the granularity one row per product ID. The IOT data is recorded per order . However product ID is a lower granularity and it can be scaled up accordingly.



Type & Attribute Descriptions of the fact and dimension Tables:

Table fact_sales

Column name	Type	Description
product_id	int	Foreign Key to join Product dimension table
date_id	int	Foreign Key to join Date dimension table
store_id	int	Foreign Key to join store dimension table
employee_id	int	Foreign Key to join employee dimension table

sales_type_id	int	Primary key of the salesfact table
TransactionNo	Varchar()	Degenerate key to record the transaction numbers
SalesAmount	float	Amount obtained from the sale
dim_weatherIOT_id	int	Foreign Key to join weather dimension table
dim_time_time_id	int	Foreign Key to join time dimension table
dim_capacityIOT_capacity_id	int	Foreign Key to join capacity dimension table
dim_NoiseIOT_NoiseIOT_id	int	Foreign Key to join Noise dimension table
dim_trafficIOT_traffic_id	int	Foreign Key to join traffic dimension table

Table dim_employee

Column name	Type	Properties	Description
employee_id	int	PK	Id of the employee
first_name	varchar(128)		First name of the employee
last_name	varchar(128)		Last name of the employee
birth_year	int		Birth year of the employee

Table dim_date

Column name	Type	Properties	Description
date_id	int	PK	
action_date	date		real date when action took place
action_week	int		ordinal number of week in year
action_month	int		actual month when action took place
action_year	int		actual month when action took place

action_weekday	varchar(16)		name of weekday when action took place (Monday, ...)

Table dim_store

Column name	Type	Properties	Description
store_id	int	PK	Id of the store
store_address	varchar(256)		Address of the store
city	varchar(128)		City in which store is located
region	varchar(128)		Region in which store is located
state	varchar(128)		State in which store is located
country	varchar(128)		Country in which store is located

Table dim_customer

Column name	Type	Properties	Description
Customer_id	int	PK	Id of the Customer
Customer_firstname	varchar(128)		First name of the Customer
Customer_lastname	varchar(128)		Last name of the Customer
Customer_gender	varchar(128)		Gender of the Customer
Customer_email	varchar(128)		Email details for the Customer
Customer_contact	bigserial		Contact details of the customer

Table dim_product

Column name	Type	Properties	Description
product_id	int	PK	Identification number of the product

product_name	varchar(256)		Name of the product
product_type	varchar(256)		Type of product

Four additional dimension tables were created in order to capture IOT data. Namely Weather, capacity, noise and time

2.6. Table dim_weatherIOT

2.6.1. dimension description

Column name	Type	Properties	Description
humidity_category	Varchar()		Humidity category value
cloud_cover_category	Varchar()		Cloud cover category value
weather_condition	Varchar()		Weather condition category
pollen_category	Varchar()		Pollen category
weather_id	int	PK	primary key for the weather dimension table
temprature_category	Varchar()		Category of the temprature

The weather dimension table was created by combining all IOT information on the weather.
This involved humidity, weather condition, pollen, cloud cover, and temperature. The first task was to ensure that all individual attributes in the table were binned into categories to bring uniformity.

Weather condition and pollen rating have a total of 10 possible values values. Temprature can have values ranging from -50.00 F to 150.00 F. This can be categorized into into 10 different distinct intervals.

Temprature_category(Value)	Bin(degree F)(Decimals excluded)
Level 1	-50F to -30F
Level 2	-30F to -10F
Level 3	-10F to - 10F
Level 4	10F to 30F
Level 5	30F to 50F
Level 6	50F to 70F
Level 7	70F to 90F
Level 8	90F to 110F
Level 9	110F to 130F

Level 10	130F to 150F

In a similar humidity and cloud_cover attributes were converted into 10 categories

humidity_category(Value)	**Bin(%)(Decimals included i.e 10 = 10.00)**
Level 1	0 to 10
Level 2	10 to 20
Level 3	20 to 30
Level 4	30 to 40
Level 5	40 to 50
Level 6	50 to 60
Level 7	60 to 70
Level 8	70 to 80
Level 9	80 to 90
Level 10	90 to 100

Cloud_cover(Value)	**Bin(%)(Decimals included i.e 10 = 10.00)**
Level 1	0 to 10
Level 2	10 to 20
Level 3	20 to 30
Level 4	30 to 40
Level 5	40 to 50
Level 6	50 to 60
Level 7	60 to 70
Level 8	70 to 80
Level 9	80 to 90
Level 10	90 to 100

2.7. Table dim_time

2.7.1. Columns

Column name	**Type**	**Description**	**Range**
hours	time	Records hours.	This column can have values ranging from 6:00 to 1:00am(19 Values)
minutes	time	Records minutes.	This column indicates minutes and can have values from 0 to 60
seconds	time	Records seconds	This column indicates minutes

			and can have values from 0 to 60
id	int	Primary key of the time dimension	

To ensure that the dimension table does not increase drastically with respect to the fact table a separate time dimension was used.

2.8. Table dim_NoiseIOT

2.8.1. Columns

Column name	Type	Properties	Description
IndoorNoiseLevel_Category	Varchar()		Indoor Noise category value
NoiseIOT_id	int	PK	Primary key for the noise table
outdoorNoiseLevel_category	Varchar()		Outdoor Noise category value

The noise dimension table was created by grouping indoor and outdoor noise data gathered by the IOT system. As with the weather IOT data the noise levels were also broken into 14 different categories with intervals of 10.

IndoorNoiselevel_category(Value)	Bin(Decimals included i.e 10 = 10.00)
Level 1	0 to 10
Level 2	10 to 20
Level 3	20 to 30
Level 4	30 to 40
Level 5	40 to 50
Level 6	50 to 60
Level 7	60 to 70
Level 8	70 to 80
Level 9	80 to 90
Level 10	90 to 100
Level 11	100 to 110
Level 12	110 to 120
Level 13	120 to 130
Level 14	130 to 140

OutdoorNoiselevel_category(Value)	Bin(Decimals included i.e 10 = 10.00)
Level 1	0 to 10
Level 2	10 to 20
Level 3	20 to 30
Level 4	30 to 40
Level 5	40 to 50
Level 6	50 to 60
Level 7	60 to 70
Level 8	70 to 80
Level 9	80 to 90
Level 10	90 to 100
Level 11	100 to 110
Level 12	110 to 120
Level 13	120 to 130
Level 14	130 to 140

2.10. Table dim_trafficIOT

2.10.1. Columns

Column name	Type	Properties	Description
traffic_category	Varchar()		Traffic category value
foot_traffic_category	Varchar()		Foot traffic category value
traffic_id	int	PK	Primary key for the traffic table

The following bins were used to categorize traffic and foot traffic

Traffic Condition	
Category	**Bins**
Extemely dense	7778 to 10,000
Very dense	5556 to 7777
dense	3334 to 5555
Moderately dense	1112 to 3333
medium	- 1112 to 1111
Lower medium dense	- 3334 to -1111
Less dense	- 5556 to -3333
Very less dense	-7778 to -5555
Extemely Less dense	-10,000 to -7777

Foot Traffic	
Category	**Bins**
Extremely High	8,001 - 10,000
High	6,001 - 8,000
Medium	4,001 - 6,000
Less	2,001 - 4,000
Very Less	0 - 2,000

2.11. Table dim_capacityIOT

2.11.1. Columns

Column name	Type	Properties	Description
Queue Length_category	Varchar()		Queue category
SeatingAvailabilityCategory	Varchar()		Category of seating Availability
capacity_id	int	PK	

Queue Length_category (Value)	Bin
category 1	1 to 5
category 2	5 to 10
category 3	10 to 15
category 4	15 to 20
category 5	20 to 25
category 6	25 to 30
category 7	30 to 35
category 8	35 to 40
category 9	40 to 45
category 10	45 to 50

SeatingAvailabilityCategory (Value)	Bin(%)
category 1	0 to 10
category 2	10 to 20
category 3	20 to 30
category 4	30 to 40
category 5	40 to 50
category 6	50 to 60
category 7	60 to 70

category 8	70 to 80
category 9	80 to 90
category 10	90 to 100

Size Calculation:

The new dimension tables weather, capacity, traffic and noise have a fixed range of values and Hence the rows will not increase for each individual transaction.

The table sizes have been computed as follows:

Weather:

Column name	Type	Size
humidity_category	Varchar()	10
cloud_cover_category	Varchar()	10
weather_condition	Varchar()	10
pollen_category	Varchar()	10
weather_id	int	4
temperature_category	Varchar()	10

Categories for temperature = 10
Categories for humidity = 10
Categories for cloud cover = 10
Categories for pollen = 10
Categories for weather = 10
Total no of Table Size = categories*Sum of sizes
= (10*10*10*10*10) * 54
= 5400000 bytes

Time:

No of Category for a Time in a day from 6.00 am to 1:00 am = 19*60*60 = 68400 Rows

No of Bytes =5

Total no of TableSize =68400*5 =342000b

Capacity:

Column name	Type	Size
Queue Length_category	int	12
SeatingAvailabilityCategory	int	12
capacity_id	int	4

Categories for queue = 10
Categories for Seating = 10

Total no of Table Size = categories*Sum of sizes
= (10*10) * 28
= 2800 bytes

Traffic:

Column name	Type	Size
traffic_category	Varchar()	20
foot_traffic_category	Varchar()	20
traffic_id	int	4

Categories for traffic = 9
Categories for foot traffic = 5

Total no of Table Size = categories*Sum of sizes
= 45 * 44 = 1980 bytes

Noise:

Column name	Type	SIze
IndoorNoiseLevel_Category	Varchar()	10

NoiseIOT_id	int	4
outdoorNoiseLevel_category	Varchar()	10

Categories for Indoor Noise Level = 14
Categories for Outdoor Noise Level = 14

Total no of Table Size = categories*Sum of sizes
= 14 *14* 24 = 4704 bytes

Sample Reports :

a. How do sales vary across stores with varying cloud cover during the hours of 8:00am – 4:00pm and Months of May – Sep.

Sample report1:

Description	This can be achieved by using the cloud cover as the column and store name as the rows. The time frames become the filter and the Sales Amount becomes the measure value
Column:	Cloud_cover_category
Row:	store_name
Filters:	DateMonth (May-Sep), Time(8:00am-4pm)
Cell values:	Sales Amount

b. How do sales vary across products with varying temperatures during the hours of 12:00pm – 5:00pm during the months of March – Sep.

Sample Report2:

Description	This can be achieved by using the temperature category as the column and product name as the rows. The time frames become the filter and the Sales Amount becomes the measure value
Column:	Temperature_category
Row:	product_name
Filters:	DateMonth (March-Sep), Time(12:00pm-5pm)
Cell values:	Sales Amount

Estimates for size of raw IoT data for 1 year:

The IOT data generated and the sizes can be summarized as below:

IOT Data	Range	Data Type	Size
Queue Length	1 to 50	Integer	4
Temperature	50.00 to 150.00	Float	4
Time	6:00:00AM to 1:00:00AM	Time	8
Humidity	0.00 to 100.00	Float	4
Cloud Cover	0.00 to 100.00	Float	4
Weather condition	10 Category	VarChar	12
Pollen	1 to 10	Integer	4
Traffic Conditions	-10,000 to +10,000	Integer	4
Seating Availability	0 to 100%	Integer	4
Foot Traffic Score	0 to 10000	Integer	4
Indoor Noise level	0.00 to 140.00	Float	4
Outdoor Noise level	0.00 to 140.00	Float	4

Assuming that there are 10000 transactions and 1 row per transaction, we can calculate the size of raw IoT data for 1year = 10,000 * 365 * 60 = 219,000,000 Bytes (60 is the total size)

This translates to :

Total Size in KB = 219,000,000/1024 = 213867.1875 KB
Total Size in MB = 213867.1875/1024 = 208.8547 MB
Total Size in GB= 208.8547/1024 = 0.20396 GB

Estimated cost of raw data storage in relational database vs object storage in cloud:

Azure SQL data storage can be used in this case for storage in relational format.

In addition to this Azure provides a 12 month free access to the SQL database and provides free 200$ credit.

A general purpose system would suffice for our needs since our data is small. The price chart can be seen below.

Gen 4

Gen 4 CPUs are based on Intel E5-2673 v3 (Haswell) 2.4 GHz processors. In Gen 4, 1 vCore = 1 physical CPU

vCORE	MEMORY (GB)	INCLUDED STORAGE	LICENSE INCLUDED PRICE[2]	AZURE HYBRID BENEFIT[1,2] PRICE (% SAVINGS)	1 YEAR RESERVED LICENSE INCLUDED PRICE (% SAVINGS)
8	56	First 32 GB/month	$2.0175/hour	$1.2178/hour (~40%)	$1.5915/hour (~21%)

Object Storage: