Datasets and Benchmark Planning

# Datasets

This section contains potential datasets to be used in benchmarking.

## Real-World

This section contains potential datasets of real-world data which can be used. They may also serve as a proof-of-concept when coming from an area where skyline queries are potentially used.

### Airbnb

Dataset that contains various real-world datasets for various cities around the world.

An efficient parallel processing method for skyline queries in MapReduce:  
Airbnb is a global online platform that hosts 150 million users and provides the lease and rental service of accommodations. The Airbnb dataset has over 9 million tuples (items) in 113 cities. Each tuple contains 21 attributes: room id, city name, price, capacity for guests, the number of reviews, overall satisfaction, the number of bedrooms, the number of bathrooms, the minimum stay for a visit, and so on.

<http://insideairbnb.com/get-the-data.html>

### Vehicle Fuel Economy

This dataset contains the fuel economy of various vehicles throughout the years.

<https://www.fueleconomy.gov/feg/download.shtml>

### CoIL 2000

Insurance dataset commonly used in benchmarks.

<https://archive.ics.uci.edu/ml/datasets/Insurance+Company+Benchmark+(COIL+2000)>

### NBA

Datasets that contain data about players and teams in the NBA (<https://www.nba.com/stats/>).

*NO OFFICIAL DATASOURCE DOWNLOAD FOUND, MANUAL EXTRACTION NECESSARY*

## Synthetic

This section contains the synthetically generated datasets which can be either independent or correlated.

### TPC-DS

<http://tpc.org/default5.asp>

System for generating data and queries. Only query generation useful for skyline query benchmarking.

### DSB: A Decision Support Benchmark for Workload-Driven and Traditional Database Systems

<https://github.com/microsoft/dsb>

Successor of TPC-DS which it is based on. Offers more versatile data generation including correlation and independent data.

# General Benchmark Planning

In this section, we discuss what is being benchmarked and what such queries may look like.

It is formulated in such a way that the query planning is independent of the actual selected set.

## Types of Benchmarks

In this section, we discuss the types of benchmarks to be undertaken detached from the actual algorithms and queries used.

We state what is plotted on which axis in the same plot and which parameters are fixed for the entire plot. If one of the fixed values is different, then this belongs to a separate plot. We also state how the data is grouped, i.e., which “lines” in the graph are plotted.

### Dimensionality vs. Execution Time

Plot how the execution time changes when more dimensions are used on a query with no changes sans the dimensions. Input size also stays the same.

|  |  |
| --- | --- |
| X-Axis | No. of dimensions |
| Y-Axis | Execution Time |
| Group By | Algorithm |
| Fixed | Query (sans dimensions) |
| Input Size |
| No. of Nodes |
| Dataset |

### Dataset Size vs. Execution Time

Plot how the execution time changes when the size of the dataset is increased.

|  |  |
| --- | --- |
| X-Axis | Input Size |
| Y-Axis | Execution Time |
| Group By | Algorithm |
| Fixed | Query |
| Dimensions |
| No. of Nodes |
| Dataset |

### Number of Nodes vs. Execution Time

Plot how the execution time changes when the number of partitions changes.

|  |  |
| --- | --- |
| X-Axis | No. of Nodes |
| Y-Axis | Execution Time |
| Group By | Algorithm |
| Fixed | Query |
| Dimensions |
| Input Size |
| Dataset |

# Algorithms

* Rewritten Skyline Query
* Block-Nested-Loop (BNL)
* Distributed BNL
* Incomplete BNL

# Test System

LBD Cluster

# Test Data and Queries

In this section, we list the order of dimensions in which they are used as “relevant” skyline dimensions and which columns correspond to the key for each dataset.

We also give skyline queries which correspond to the given dimensions and their equivalent in “plain” SQL without the specialized skyline syntax.

## Vehicle Fuel Economy (vehicles.csv)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Col | Property | Distinct Values | Type | Description |
|  | **IDs** | | | |
| 1 | make | 134 | N/A | Make of the car |
| 2 | model | 4399 | N/A | Model of the car |
|  | **Skyline Dimensions** | | | |
| 3 | fuelCost08 | 69 | MIN | annual fuel cost |
| 4 | barrels06 | 232 | MIN | annual petroleum consumption in barrels |
| 5 | city08 | 52 | MAX | city MPG |
| 6 | highway08 | 52 | MAX | highway MPG |
| 7 | comb08 | 50 | MAX | combined MPG |
| 8 | combinedCD | 10 | MIN | combined gasoline consumption (gallons/100 miles) in charge depleting mode |
| 9 | displ | 65 | MAX | engine displacement (liters) |
| 10 | cylinders | 9 | MAX | number of cylinders |
| 11 | cityUF | 117 | MAX | City Utility Factor - Fraction of vehicle miles traveled under electricity (Plug-In-Hybrids) |

### 1 dimension

Skyline Syntax:

|  |
| --- |
| SELECT \* FROM vehicles SKYLINE OF  fuelCost08 MIN |

“Plain” SQL:

|  |
| --- |
| SELECT \* FROM vehicles AS o WHERE NOT EXISTS(  SELECT \* FROM vehicles AS i WHERE  i.fuelCost08 <= o.fuelCost08  AND (  i.fueCost08 < o.fuelCost08  ) ) |

### 2 dimensions

Skyline Syntax:

|  |
| --- |
| SELECT \* FROM vehicles SKYLINE OF  fuelCost08 MIN,  barrels06 MIN |

“Plain” SQL:

|  |
| --- |
| SELECT \* FROM vehicles AS o WHERE NOT EXISTS(  SELECT \* FROM vehicles AS i WHERE  i.fuelCost08 <= o.fuelCost08 AND  i.barrels06 <= o.barrels06  AND (  i.fueCost08 < o.fuelCost08 OR  i.barrels06 < o.barrels06  ) ) |

### 3 dimensions

Skyline Syntax:

|  |
| --- |
| SELECT \* FROM vehicles SKYLINE OF  fuelCost08 MIN,  barrels06 MIN,  city08 MAX |

“Plain” SQL:

|  |
| --- |
| SELECT \* FROM vehicles AS o WHERE NOT EXISTS(  SELECT \* FROM vehicles AS i WHERE  i.fuelCost08 <= o.fuelCost08 AND  i.barrels06 <= o.barrels06 AND  i.city08 >= o.city08  AND (  i.fueCost08 < o.fuelCost08 OR  i.barrels06 < o.barrels06 OR  i.city08 > o.city08  ) ) |

### 4 dimensions

Skyline Syntax:

|  |
| --- |
| SELECT \* FROM vehicles SKYLINE OF  fuelCost08 MIN,  barrels06 MIN,  city08 MAX,  highway08 MAX |

“Plain” SQL:

|  |
| --- |
| SELECT \* FROM vehicles AS o WHERE NOT EXISTS(  SELECT \* FROM vehicles AS i WHERE  i.fuelCost08 <= o.fuelCost08 AND  i.barrels06 <= o.barrels06 AND  i.city08 >= o.city08 AND  i.highway08 >= o.highway08  AND (  i.fueCost08 < o.fuelCost08 OR  i.barrels06 < o.barrels06 OR  i.city08 > o.city08 OR  i.highway08 > o.highway08  ) ) |

### 5 dimensions

Skyline Syntax:

|  |
| --- |
| SELECT \* FROM vehicles SKYLINE OF  fuelCost08 MIN,  barrels06 MIN,  city08 MAX,  highway08 MAX,  comb08 MAX |

“Plain” SQL:

|  |
| --- |
| SELECT \* FROM vehicles AS o WHERE NOT EXISTS(  SELECT \* FROM vehicles AS i WHERE  i.fuelCost08 <= o.fuelCost08 AND  i.barrels06 <= o.barrels06 AND  i.city08 >= o.city08 AND  i.highway08 >= o.highway08 AND  i.comb08 >= o.comb08  AND (  i.fueCost08 < o.fuelCost08 OR  i.barrels06 < o.barrels06 OR  i.city08 > o.city08 OR  i.highway08 > o.highway08 OR  i.comb08 > o.comb08  ) ) |

### 6 dimensions

Skyline Syntax:

|  |
| --- |
| SELECT \* FROM vehicles SKYLINE OF  fuelCost08 MIN,  barrels06 MIN,  city08 MAX,  highway08 MAX,  comb08 MAX,  combinedCD MIN |

“Plain” SQL:

|  |
| --- |
| SELECT \* FROM vehicles AS o WHERE NOT EXISTS(  SELECT \* FROM vehicles AS i WHERE  i.fuelCost08 <= o.fuelCost08 AND  i.barrels06 <= o.barrels06 AND  i.city08 >= o.city08 AND  i.highway08 >= o.highway08 AND  i.comb08 >= o.comb08 AND  i.combinedCD <= o.combinedCD  AND (  i.fueCost08 < o.fuelCost08 OR  i.barrels06 < o.barrels06 OR  i.city08 > o.city08 OR  i.highway08 > o.highway08 OR  i.comb08 > o.comb08 OR  i.combinedCD < o.combinedCD  ) ) |

## Inside Airbnb

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Col | Property | Distinct Values | Type | Description |
|  | **IDs** | | | |
| 1 | id | 819455 | N/A | global unique id of room |
|  | **Skyline Dimensions** | | | |
| 2 | price | 8912 | MIN | price per night |
| 3 | accommodates | 16 | MAX | max number of people accommodated |
| 4 | bedrooms | 39 | MAX | number of bedrooms |
| 5 | beds | 62 | MAX | number of beds |
| 6 | number\_of\_reviews | 814 | MAX | number of reviews |
| 7 | review\_scores\_rating | 216 | MAX | review score for rating |
| 8 | review\_scores\_accuracy | 206 | MAX | review score for accuracy |
| 9 | review\_scores\_cleanliness | 238 | MAX | review score for cleanliness |
| 10 | review\_scores\_checkin | 203 | MAX | review score for check-in |
| 11 | review\_scores\_communication | 201 | MAX | review score for communication (with host) |
| 12 | review\_scores\_location | 214 | MAX | review score for location |
| 13 | review\_scores\_value | 214 | MAX | review score for value |

### 1 dimension

Skyline Syntax:

|  |
| --- |
| SELECT \* FROM airbnb SKYLINE OF  price MIN |

“Plain” SQL:

|  |
| --- |
| SELECT \* FROM airbnb AS o WHERE NOT EXISTS(  SELECT \* FROM airbnb AS i WHERE  i.price <= o.price  AND (  i.price < o.price  ) ) |

### 2 dimensions

Skyline Syntax:

|  |
| --- |
| SELECT \* FROM airbnb SKYLINE OF  price MIN,  accommodates MAX |

“Plain” SQL:

|  |
| --- |
| SELECT \* FROM airbnb AS o WHERE NOT EXISTS(  SELECT \* FROM airbnb AS i WHERE  i.price <= o.price AND  i.accomodates >= o.accomodates  AND (  i.price < o.price OR  i.accomodates > o.accomodates  ) ) |

### 3 dimensions

Skyline Syntax:

|  |
| --- |
| SELECT \* FROM airbnb SKYLINE OF  price MIN,  accommodates MAX,  bedrooms MAX |

“Plain” SQL:

|  |
| --- |
| SELECT \* FROM airbnb AS o WHERE NOT EXISTS(  SELECT \* FROM airbnb AS i WHERE  i.price <= o.price AND  i.accomodates >= o.accomodates AND  i.bedrooms >= o.bedrooms  AND (  i.price < o.price OR  i.accomodates > o.accomodates OR  i.bedrooms > o.bedrooms  ) ) |

### 4 dimensions

Skyline Syntax:

|  |
| --- |
| SELECT \* FROM airbnb SKYLINE OF  price MIN,  accommodates MAX,  bedrooms MAX,  beds MAX |

“Plain” SQL:

|  |
| --- |
| SELECT \* FROM airbnb AS o WHERE NOT EXISTS(  SELECT \* FROM airbnb AS i WHERE  i.price <= o.price AND  i.accomodates >= o.accomodates AND  i.bedrooms >= o.bedrooms AND  i.beds >= o.beds  AND (  i.price < o.price OR  i.accomodates > o.accomodates OR  i.bedrooms > o.bedrooms OR  i.beds > o.beds  ) ) |

### 5 dimensions

Skyline Syntax:

|  |
| --- |
| SELECT \* FROM airbnb SKYLINE OF  price MIN,  accommodates MAX,  bedrooms MAX,  beds MAX,  number\_of\_reviews MAX |

“Plain” SQL:

|  |
| --- |
| SELECT \* FROM airbnb AS o WHERE NOT EXISTS(  SELECT \* FROM airbnb AS i WHERE  i.price <= o.price AND  i.accomodates >= o.accomodates AND  i.bedrooms >= o.bedrooms AND  i.beds >= o.beds AND  i.number\_of\_reviews >= o.number\_of\_reviews  AND (  i.price < o.price OR  i.accomodates > o.accomodates OR  i.bedrooms > o.bedrooms OR  i.beds > o.beds OR  i.number\_of\_reviews > o.number\_of\_reviews  ) ) |

### 6 dimensions

Skyline Syntax:

|  |  |
| --- | --- |
| SELECT \* FROM airbnb SKYLINE OF  price MIN,  accommodates MAX,  bedrooms MAX,  beds MAX,  number\_of\_reviews MAX,  review\_scores\_rating MAX |  |

“Plain” SQL:

|  |  |
| --- | --- |
| SELECT \* FROM airbnb AS o WHERE NOT EXISTS(  SELECT \* FROM airbnb AS i WHERE  i.price <= o.price AND  i.accomodates >= o.accomodates AND  i.bedrooms >= o.bedrooms AND  i.beds >= o.beds AND  i.number\_of\_reviews >= o.number\_of\_reviews AND  i.review\_scores\_rating >= o.review\_scores\_rating  AND (  i.price < o.price OR  i.accomodates > o.accomodates OR  i.bedrooms > o.bedrooms OR  i.beds > o.beds OR  i.number\_of\_reviews > o.number\_of\_reviews OR  i.review\_scores\_rating > o.review\_scores\_rating  ) ) |  |

## NBA

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Col | Property | Distinct Values | Type | Description |
|  | **IDs** | | | |
| 1 | PLAYER\_ID | 446 | N/A | Unique Player ID |
|  | **Skyline Dimensions** | | | |
| 2 | W | 10 | MAX | Number of Wins |
| 3 | L | 11 | MIN | Number of Losses |
| 4 | W\_PCT | 40 | MAX | Win percentage |
| 5 | FGM | 96 | MAX | Field Goals Made |
| 6 | FGA | 166 | MAX | Field Goals Attempted |
| 7 | FG\_PCT | 222 | MAX | Field Goal Percentage |
| 8 | FTM | 53 | MAX | Free Throws Made |
| 9 | FTA | 63 | MAX | Free Throws Attempted |
| 10 | OREB | 38 | MAX | Offensive Rebounds |
| 11 | DREB | 85 | MAX | Defensive Rebounds |
| 12 | REB | 101 | MAX | Rebounds (total) |
| 13 | AST | 71 | MAX | Assists |

### 1 dimension

Skyline Syntax:

|  |
| --- |
| SELECT \* FROM nba SKYLINE OF  W MAX |

“Plain” SQL:

|  |
| --- |
| SELECT \* FROM nba AS o WHERE NOT EXISTS(  SELECT \* FROM nba AS i WHERE  i.W >= o.W  AND (  i.W > o.W  ) ) |

### 2 dimensions

Skyline Syntax:

|  |
| --- |
| SELECT \* FROM nba SKYLINE OF  W MAX,  L MIN |

“Plain” SQL:

|  |
| --- |
| SELECT \* FROM nba AS o WHERE NOT EXISTS(  SELECT \* FROM nba AS i WHERE  i.W >= o.W AND  i.L <= o.L  AND (  i.W > o.W OR  i.L < o.L  ) ) |

### 3 dimensions

Skyline Syntax:

|  |
| --- |
| SELECT \* FROM nba SKYLINE OF  W MAX,  L MIN,  W\_PCT MAX |

“Plain” SQL:

|  |
| --- |
| SELECT \* FROM nba AS o WHERE NOT EXISTS(  SELECT \* FROM nba AS i WHERE  i.W >= o.W AND  i.L <= o.L AND  i.W\_PCT >= o.W\_PCT  AND (  i.W > o.W OR  i.L < o.L OR  i.W\_PCT > o.W\_PCT  ) ) |

### 4 dimensions

Skyline Syntax:

|  |
| --- |
| SELECT \* FROM nba SKYLINE OF  W MAX,  L MIN,  W\_PCT MAX,  FGM MAX |

“Plain” SQL:

|  |
| --- |
| SELECT \* FROM nba AS o WHERE NOT EXISTS(  SELECT \* FROM nba AS i WHERE  i.W >= o.W AND  i.L <= o.L AND  i.W\_PCT >= o.W\_PCT AND  i.FGM >= o.FGM  AND (  i.W > o.W OR  i.L < o.L OR  i.W\_PCT > o.W\_PCT OR  i.FGM > o.FGM  ) ) |

### 5 dimensions

Skyline Syntax:

|  |
| --- |
| SELECT \* FROM nba SKYLINE OF  W MAX,  L MIN,  W\_PCT MAX,  FGM MAX,  FGA MAX |

“Plain” SQL:

|  |
| --- |
| SELECT \* FROM nba AS o WHERE NOT EXISTS(  SELECT \* FROM nba AS i WHERE  i.W >= o.W AND  i.L <= o.L AND  i.W\_PCT >= o.W\_PCT AND  i.FGM >= o.FGM AND  i.FGA >= o.FGA  AND (  i.W > o.W OR  i.L < o.L OR  i.W\_PCT > o.W\_PCT OR  i.FGM > o.FGM OR  i.FGA > o.FGA  ) ) |

### 6 dimensions

Skyline Syntax:

|  |  |
| --- | --- |
| SELECT \* FROM nba SKYLINE OF  W MAX,  L MIN,  W\_PCT MAX,  FGM MAX,  FGA MAX,  FG\_PCT MAX |  |

“Plain” SQL:

|  |  |
| --- | --- |
| SELECT \* FROM nba AS o WHERE NOT EXISTS(  SELECT \* FROM nba AS i WHERE  i.W >= o.W AND  i.L <= o.L AND  i.W\_PCT >= o.W\_PCT AND  i.FGM >= o.FGM AND  i.FGA >= o.FGA AND  i.FG\_PCT >= o.FG\_PCT  AND (  i.W > o.W OR  i.L < o.L OR  i.W\_PCT > o.W\_PCT OR  i.FGM > o.FGM OR  i.FGA > o.FGA OR  i.FG\_PCT > o.FG\_PCT  ) ) |  |

## COIL 2000

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Col | Property | Distinct Values | Type | Description |
|  | **IDs** | | | |
| 1 | ID | 5822 | N/A | Unique ID |
|  | **Skyline Dimensions** | | | |
| 2 | MOSHOOFD | 10 | MAX | Customer Main Type |
| 3 | MGODRK | 10 | MIN | Roman catholic |
| 4 | MGODPR | 10 | MIN | Protestant |
| 5 | MGODGE | 10 | MAX | No religion |
| 6 | MRELGE | 10 | MAX | Married |
| 7 | MRELOV | 10 | MAX | Other relation |
| 8 | MFALLEEN | 10 | MIN | Singles |
| 9 | MFGEKIND | 10 | MIN | Household without children |
| 10 | MFWEKIND | 10 | MAX | Household with children |
| 11 | MOPLHOOG | 10 | MAX | High level education |
| 12 | MOPLMIDD | 10 | MAX | Medium level education |
| 13 | MOPLLAAG | 10 | MIN | Lower-level education |

(MIN/MAX chosen arbitrarily)

### 1 dimension

Skyline Syntax:

|  |
| --- |
| SELECT \* FROM coil2000 SKYLINE OF  MOSHOOFD MAX |

“Plain” SQL:

|  |
| --- |
| SELECT \* FROM coil2000 AS o WHERE NOT EXISTS(  SELECT \* FROM coil2000 AS i WHERE  i.MOSHOOFD >= o.MOSHOOFD  AND (  i.MOSHOOFD > o.MOSHOOFD  ) ) |

### 2 dimensions

Skyline Syntax:

|  |
| --- |
| SELECT \* FROM coil2000 SKYLINE OF  MOSHOOFD MAX,  MGODRK MIN |

“Plain” SQL:

|  |
| --- |
| SELECT \* FROM coil2000 AS o WHERE NOT EXISTS(  SELECT \* FROM coil2000 AS i WHERE  i.MOSHOOFD >= o.MOSHOOFD AND  i.MGODRK <= o.MGODRK  AND (  i.MOSHOOFD > o.MOSHOOFD OR  i.MGODRK < o.MGODRK  ) ) |

### 3 dimensions

Skyline Syntax:

|  |
| --- |
| SELECT \* FROM coil2000 SKYLINE OF  MOSHOOFD MAX,  MGODRK MIN,  MGODPR MIN |

“Plain” SQL:

|  |
| --- |
| SELECT \* FROM coil2000 AS o WHERE NOT EXISTS(  SELECT \* FROM coil2000 AS i WHERE  i.MOSHOOFD >= o.MOSHOOFD AND  i.MGODRK <= o.MGODRK AND  i.MGODPR MIN <= o.MGODPR  AND (  i.MOSHOOFD > o.MOSHOOFD OR  i.MGODRK < o.MGODRK OR  i.MGODPR MIN < o.MGODPR  ) ) |

### 4 dimensions

Skyline Syntax:

|  |
| --- |
| SELECT \* FROM coil2000 SKYLINE OF  MOSHOOFD MAX,  MGODRK MIN,  MGODPR MIN,  MGODGE MAX |

“Plain” SQL:

|  |
| --- |
| SELECT \* FROM coil2000 AS o WHERE NOT EXISTS(  SELECT \* FROM coil2000 AS i WHERE  i.MOSHOOFD >= o.MOSHOOFD AND  i.MGODRK <= o.MGODRK AND  i.MGODPR MIN <= o.MGODPR AND  i.MGODGE >= o.MGODGE  AND (  i.MOSHOOFD > o.MOSHOOFD OR  i.MGODRK < o.MGODRK OR  i.MGODPR MIN < o.MGODPR OR  i.MGODGE > o.MGODGE  ) ) |

### 5 dimensions

Skyline Syntax:

|  |
| --- |
| SELECT \* FROM coil2000 SKYLINE OF  MOSHOOFD MAX,  MGODRK MIN,  MGODPR MIN,  MGODGE MAX,  MRELGE MAX |

“Plain” SQL:

|  |
| --- |
| SELECT \* FROM coil2000 AS o WHERE NOT EXISTS(  SELECT \* FROM coil2000 AS i WHERE  i.MOSHOOFD >= o.MOSHOOFD AND  i.MGODRK <= o.MGODRK AND  i.MGODPR MIN <= o.MGODPR AND  i.MGODGE >= o.MGODGE AND  i.MRELGE >= o.MRELGE  AND (  i.MOSHOOFD > o.MOSHOOFD OR  i.MGODRK < o.MGODRK OR  i.MGODPR MIN < o.MGODPR OR  i.MGODGE > o.MGODGE OR  i.MRELGE > o.MRELGE  ) ) |

### 6 dimensions

Skyline Syntax:

|  |
| --- |
| SELECT \* FROM coil2000 SKYLINE OF  MOSHOOFD MAX,  MGODRK MIN,  MGODPR MIN,  MGODGE MAX,  MRELGE MAX,  MRELOV MAX |

“Plain” SQL:

|  |
| --- |
| SELECT \* FROM coil2000 AS o WHERE NOT EXISTS(  SELECT \* FROM coil2000 AS i WHERE  i.MOSHOOFD >= o.MOSHOOFD AND  i.MGODRK <= o.MGODRK AND  i.MGODPR MIN <= o.MGODPR AND  i.MGODGE >= o.MGODGE AND  i.MRELGE >= o.MRELGE AND  i.MRELOV >= o.MRELOV  AND (  i.MOSHOOFD > o.MOSHOOFD OR  i.MGODRK < o.MGODRK OR  i.MGODPR MIN < o.MGODPR OR  i.MGODGE > o.MGODGE OR  i.MRELGE > o.MRELGE OR  i.MRELOV > o.MRELOV  ) ) |

### DSB Benchmarking Dataset

We use the store sales for benchmarking.



(Taken from the specification provided on GitHub)

| Column | Datatype | NULL | Primary Key | Foreign Key |
| --- | --- | --- | --- | --- |
| ss\_sold\_date\_sk | identifier |  |  | d\_date\_sk |
| ss\_sold\_time\_sk | identifier |  |  | t\_time\_sk |
| ss\_item\_sk | identifier | N | Y | i\_item\_sk |
| ss\_customer\_sk | identifier |  |  | c\_customer\_sk |
| ss\_cdemo\_sk | identifier |  |  | cd\_demo\_sk |
| ss\_hdemo\_sk | identifier |  |  | hd\_demo\_sk |
| ss\_addr\_sk | identifier |  |  | ca\_address\_sk |
| ss\_store\_sk | identifier |  |  | s\_store\_sk |
| ss\_promo\_sk | identifier |  |  | p\_promo\_sk |
| ss\_ticket\_number | identifier | N | Y |  |
| ss\_quantity | integer |  |  |  |
| ss\_wholesale\_cost | decimal(7,2) |  |  |  |
| ss\_list\_price | decimal(7,2) |  |  |  |
| ss\_sales\_price | decimal(7,2) |  |  |  |
| ss\_ext\_discount\_amt | decimal(7,2) |  |  |  |
| ss\_ext\_sales\_price | decimal(7,2) |  |  |  |
| ss\_ext\_wholesale\_cost | decimal(7,2) |  |  |  |
| ss\_ext\_list\_price | decimal(7,2) |  |  |  |
| ss\_ext\_tax | decimal(7,2) |  |  |  |
| ss\_coupon\_amt | decimal(7,2) |  |  |  |
| ss\_net\_paid | decimal(7,2) |  |  |  |
| ss\_net\_paid\_inc\_tax | decimal(7,2) |  |  |  |
| ss\_net\_profit | decimal(7,2) |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Col | Property | Type | Description |
|  | **IDs** | | |
| 1 | ss\_item\_sk | N/A | Unique ID |
| 2 | ss\_ticket\_number | N/A | Unique ID |
|  | **Skyline Dimensions** | | |
| 3 | ss\_quantity | MAX |  |
| 4 | ss\_wholesale\_cost | MIN |  |
| 5 | ss\_list\_price | MIN |  |
| 6 | ss\_sales\_price | MIN |  |
| 7 | ss\_ext\_discount\_amt | MAX |  |
| 8 | ss\_ext\_sales\_price | MIN |  |
| 9 | ss\_ext\_wholesale\_cost | MIN |  |
| 10 | ss\_ext\_list\_price | MIN |  |
| 11 | ss\_ext\_tax | MIN |  |
| 12 | ss\_coupon\_amt | MAX |  |
| 13 | ss\_ext\_sales\_price | MIN |  |
| 14 | ss\_ext\_wholesale\_cost | MIN |  |

### 1 dimension

Skyline Syntax:

|  |
| --- |
| SELECT \* FROM store\_sales SKYLINE OF  ss\_quantity MAX |

“Plain” SQL:

|  |
| --- |
| SELECT \* FROM store\_sales AS o WHERE NOT EXISTS(  SELECT \* FROM store\_sales AS i WHERE  i.ss\_quantity >= o.ss\_quantity  AND (  i.ss\_quantity > o.ss\_quantity  ) ) |

### 2 dimensions

Skyline Syntax:

|  |
| --- |
| SELECT \* FROM store\_sales SKYLINE OF  ss\_quantity MAX,  ss\_wholesale\_cost MIN |

“Plain” SQL:

|  |
| --- |
| SELECT \* FROM store\_sales AS o WHERE NOT EXISTS(  SELECT \* FROM store\_sales AS i WHERE  i.ss\_quantity >= o.ss\_quantity AND  i.ss\_wholesale\_cost <= o.ss\_wholesale\_cost  AND (  i.ss\_quantity > o.ss\_quantity OR  i.ss\_wholesale\_cost < o.ss\_wholesale\_cost  ) ) |

### 3 dimensions

Skyline Syntax:

|  |
| --- |
| SELECT \* FROM store\_sales SKYLINE OF  ss\_quantity MAX,  ss\_wholesale\_cost MIN,  ss\_list\_price MIN |

“Plain” SQL:

|  |
| --- |
| SELECT \* FROM store\_sales AS o WHERE NOT EXISTS(  SELECT \* FROM store\_sales AS i WHERE  i.ss\_quantity >= o.ss\_quantity AND  i.ss\_wholesale\_cost <= o.ss\_wholesale\_cost AND  i.ss\_list\_price <= o.ss\_list\_price  AND (  i.ss\_quantity > o.ss\_quantity OR  i.ss\_wholesale\_cost < o.ss\_wholesale\_cost OR  i.ss\_list\_price < o.ss\_list\_price  ) ) |

### 4 dimensions

Skyline Syntax:

|  |
| --- |
| SELECT \* FROM store\_sales SKYLINE OF  ss\_quantity MAX,  ss\_wholesale\_cost MIN,  ss\_list\_price MIN,  ss\_sales\_price MIN |

“Plain” SQL:

|  |
| --- |
| SELECT \* FROM store\_sales AS o WHERE NOT EXISTS(  SELECT \* FROM store\_sales AS i WHERE  i.ss\_quantity >= o.ss\_quantity AND  i.ss\_wholesale\_cost <= o.ss\_wholesale\_cost AND  i.ss\_list\_price <= o.ss\_list\_price AND  i.ss\_sales\_price <= o.ss\_sales\_price  AND (  i.ss\_quantity > o.ss\_quantity OR  i.ss\_wholesale\_cost < o.ss\_wholesale\_cost OR  i.ss\_list\_price < o.ss\_list\_price OR  i.ss\_sales\_price < o.ss\_sales\_price  ) ) |

### 5 dimensions

Skyline Syntax:

|  |
| --- |
| SELECT \* FROM store\_sales SKYLINE OF  ss\_quantity MAX,  ss\_wholesale\_cost MIN,  ss\_list\_price MIN,  ss\_sales\_price MIN,  ss\_ext\_distcount\_atm MAX |

“Plain” SQL:

|  |
| --- |
| SELECT \* FROM store\_sales AS o WHERE NOT EXISTS(  SELECT \* FROM store\_sales AS i WHERE  i.ss\_quantity >= o.ss\_quantity AND  i.ss\_wholesale\_cost <= o.ss\_wholesale\_cost AND  i.ss\_list\_price <= o.ss\_list\_price AND  i.ss\_sales\_price <= o.ss\_sales\_price AND  i.ss\_ext\_discount\_atm >= o.ss\_ext\_discount\_atm  AND (  i.ss\_quantity > o.ss\_quantity OR  i.ss\_wholesale\_cost < o.ss\_wholesale\_cost OR  i.ss\_list\_price < o.ss\_list\_price OR  i.ss\_sales\_price < o.ss\_sales\_price OR  i.ss\_ext\_discount\_atm > o.ss\_ext\_discount\_atm  ) ) |

### 6 dimensions

Skyline Syntax:

|  |
| --- |
| SELECT \* FROM store\_sales SKYLINE OF  ss\_quantity MAX,  ss\_wholesale\_cost MIN,  ss\_list\_price MIN,  ss\_sales\_price MIN,  ss\_ext\_distcount\_atm MAX,  ss\_ext\_sales\_price MIN |

“Plain” SQL:

|  |
| --- |
| SELECT \* FROM store\_sales AS o WHERE NOT EXISTS(  SELECT \* FROM store\_sales AS i WHERE  i.ss\_quantity >= o.ss\_quantity AND  i.ss\_wholesale\_cost <= o.ss\_wholesale\_cost AND  i.ss\_list\_price <= o.ss\_list\_price AND  i.ss\_sales\_price <= o.ss\_sales\_price AND  i.ss\_ext\_discount\_atm >= o.ss\_ext\_discount\_atm AND  i.ss\_ext\_sales\_price <= o.ss\_ext\_sales\_price  AND (  i.ss\_quantity > o.ss\_quantity OR  i.ss\_wholesale\_cost < o.ss\_wholesale\_cost OR  i.ss\_list\_price < o.ss\_list\_price OR  i.ss\_sales\_price < o.ss\_sales\_price OR  i.ss\_ext\_discount\_atm > o.ss\_ext\_discount\_atm OR  i.ss\_ext\_sales\_price < o.ss\_ext\_sales\_price  ) ) |