

**FIT5137 Assignment 3 Data Cleaning Report**

**Lab 03 Monday 10:00 – 12:00**

Prepared By:

1. Wong Tsz Yung Venus 33498490 (Oracle account: S33498490)
2. Justin Wu 33489068 (Oracle account: S33489068)
3. Peichun Shih 33475881 (Oracle account: S33475881)

## 1. Data Dictionary

**Table Name: Article**

Attribute Name	Description	Data Type	Character Length/Format	Acceptable Null values(Y/N?)	Primary Key(Y/N?)
articleCode	article code	Varchar2	20	N	Y
articleName	name of article	Varchar2	50	N	N
VendorKey	vendor key	Char	3	N	N
VendorName	name of vendor	Varchar2	20	N	N
categoryInit	category ID	Number	2 digits in total 0 for decimal point	Y	N
categoryName	name of category	Varchar2	20	Y	N
TypeInit	type ID	Number	2 digits in total 0 for decimal point	Y	N
TypeName	name of the type	Varchar2	20	N	N
startDate	date of start	Date	DD-Mon-RR	Y	N
expireDate	date of expire	Date	DD-Mon-RR	Y	N
colourInit	colour ID	Number	3 digits in total 0 for decimal point	Y	N
colourName	name of the colour	Varchar2	20	Y	N
sex	sex	Char	2	Y	N
picture	picture	Varchar2	10	Y	N
basePrice	base price	Number	12 digits in total 2 for decimal point	Y	N
salePrice	sale price	Number	12 digits in total 2 for decimal point	Y	N
notes	notes	Varchar2	100	Y	N

**Table Name: Cashier**

Attribute Name	Description	Data Type	Character Length/Format	Acceptable Null values(Y/N?)	Primary Key(Y/N?)
noTrans	transaction number	Varchar2	16	N	Y
dateTrans	transaction date	Date	DD-Mon-RR	N	N
typeTrans	transaction type	Varchar2	10	Y	N
Notes	Notes	Varchar2	30	Y	N
UserID	user ID	Varchar2	20	Y	N
referenceTrans	reference transaction	Varchar2	20	Y	N

**Table Name: Inventory**

<b>Attribute Name</b>	<b>Description</b>	<b>Data Type</b>	<b>Character Length/Format</b>	<b>Acceptable Null values(Y/N?)</b>	<b>Primary Key(Y/N?)</b>
articleCode	article code	Varchar2	20	N	N
barcode	barcode	Varchar2	12	N	Y
sizes	size	Varchar2	5	N	N
CurrentBasePrice	current base price	Number	12 digits in total 2 for decimal point	Y	N
CurrentSalePrice	current sale price	Number	12 digits in total 2 for decimal point	Y	N
consignment	consignment	Number	3 digits in total; 0 for decimal point	Y	N
consignmentRp	consignment resale price	Number	12 digits in total 2 for decimal point	Y	N
qty	quantity	Number	3 digits in total; 0 for decimal point	N	N
status	status	Char	1	N	N

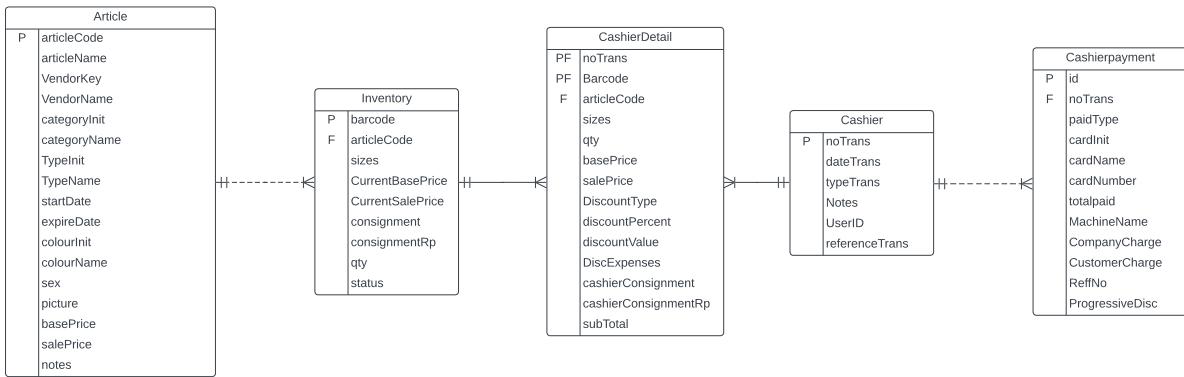
**Table Name: CashierDetail**

<b>Attribute Name</b>	<b>Description</b>	<b>Data Type</b>	<b>Character Length/Format</b>	<b>Acceptable Null values(Y/N?)</b>	<b>Primary Key(Y/N?)</b>
noTrans	transaction ID	Varchar2	16	N	Y
ArticleCode	article code	Varchar2	20	N	Y
Barcode	barcode	Varchar2	12	N	Y
sizes	size	Varchar2	5	N	N
qty	quantity	Number	3 digits in total; 0 for decimal point	N	N
basePrice	base price	Number	12 digits in total 2 for decimal point	Y	N
salePrice	sale price	Number	12 digits in total 2 for decimal point	Y	N
DiscountType	discount type	Char	1	Y	N
discountPercent	discount percentage	Number	5 digits in total 2 for decimal point	Y	N
discountValue	discount value	Number	12 digits in total 2 for decimal point	Y	N
DiscExpenses	discount availability	Char	1	Y	N
CashierConsignment	consignment	Number	3 digits in total; 0 for decimal point	Y	N
CashierConsignmentRp	consignment resale price	Number	12 digits in total 2 for decimal point	Y	N
subTotal	subtotal	Number	12 digits in total 2 for decimal point	Y	N
payment	payment	Number	12 digits in total 2 for decimal point	Y	N

**Table Name: CashierPayment**

<b>Attribute Name</b>	<b>Description</b>	<b>Data Type</b>	<b>Character Length/Format</b>	<b>Acceptable Null values(Y/N?)</b>	<b>Primary Key(Y/N?)</b>
id	cashier ID	Char	5	N	Y
noTrans	transaction ID	Varchar2	16	N	N
paidType	type of payment	Varchar2	20	N	N
cardInit	card ID	Char	3	Y	N
cardName	card name	Varchar2	20	Y	N
cardNumber	card number	Varchar2	20	Y	N
totalpaid	total amount paid	Number	12 digits in total 2 for decimal point	Y	N
MachineName	name of the machine	Char	3	Y	N
CompanyCharge	charge to company	Number	12 digits in total 2 for decimal point	Y	N
CustomerCharge	charge to customer	Number	12 digits in total 2 for decimal point	Y	N
ReffNo	reference number	Varchar2	10	Y	N
ProgressiveDisc	progressive discount	Varchar2	10	N	N

## 2. ERD Diagram



## 3. Data importing and cleaning strategies

We imported data from raw data csv files, and for rows containing errors, we manually add the correct data to the insert statement to prevent data loss.

After importing data, we cleaned the data by checking various conditions:

### 1. Duplication of data

We checked whether there is duplication of record using group by function for primary key(s) for all tables. Duplication records, if found, will be deleted.

### 2. Error in Relationship or Constraints

For tables with foreign key(s) from other table(s), we checked whether it is valid, which the data in foreign key field should exist in the primary key field of the parent table. If invalid foreign key values are found, those records will be deleted.

### 3. Inconsistent Values

We performed data cleaning for applicable attributes of all tables, checking their consistency. We considered data to be inconsistent if the values are different in units, codes, precision, language, and format.

Inconsistent values are updated with appropriate values once found. The appropriate values are determined by methods like mean, median, mode and neighboring values. As data cleaning for different fields differs, the method used for each field will be stated in the error log.

### 4. Incorrect Values

We also performed data cleaning for applicable attributes of all tables, checking their correctness. We considered data to be incorrect if it has incorrect spelling, not within a correct data range, illogical values, or incorrect data type.

Incorrect values are updated with correct values once found. The correct values are determined by methods like mean, median, mode, neighboring values, and formulas. As the correct values are computed differently for each field, method used for each field will be illustrated in error log.

## 5. Null Records

There are some fields that should not be null, we checked against it to ensure there is no empty value in that field. When null values are found, appropriate value will be assigned to it. We determine appropriate value by mean, mode, median, neighboring values, or formula. As the appropriate values are different for each field, method used for each field will be illustrated in error log.

## 5. DDL Scripts for Creating Tables

```
16 | /* Article */
17 | CREATE TABLE article (
18 |   articlecode VARCHAR2(16) NOT NULL,
19 |   articlename VARCHAR2(50) NOT NULL,
20 |   vendorkey CHAR(3) NOT NULL,
21 |   vendorname VARCHAR2(20) NOT NULL,
22 |   categoryinit NUMBER(2, 0),
23 |   categoryname VARCHAR2(20),
24 |   typeinit NUMBER(2, 0),
25 |   typename VARCHAR2(20) NOT NULL,
26 |   startdate DATE,
27 |   expiredate DATE,
28 |   colourinit NUMBER(3, 0),
29 |   colourname VARCHAR2(20),
30 |   sex CHAR(2),
31 |   picture VARCHAR2(10),
32 |   baseprice NUMBER(12, 2),
33 |   saleprice NUMBER(12, 2),
34 |   notes VARCHAR2(100),
35 |   CONSTRAINT pk_article PRIMARY KEY ( articlecode ),
36 |   CONSTRAINT ck_sex CHECK ( sex IN ( 'F', 'KD', 'M', 'U' ) ),
37 |   CONSTRAINT ck_date CHECK ( startdate < expiredate )
38 | );
39 |
40 | COMMENT ON COLUMN article.articlecode IS
41 |   'Article Code';
42 |
43 | COMMENT ON COLUMN article.articlename IS
44 |   'Name of Article';
45 |
46 | COMMENT ON COLUMN article.vendorkey IS
47 |   'Vendor Key';
48 |
49 | COMMENT ON COLUMN article.vendorname IS
50 |   'Name of Vendor';
51 |
52 | COMMENT ON COLUMN article.categoryinit IS
53 |   'Category ID';
54 |
55 | COMMENT ON COLUMN article.categoryname IS
56 |   'Name of Category';
57 |
58 | COMMENT ON COLUMN article.typeinit IS
59 |   'Type ID';
60 |
61 | COMMENT ON COLUMN article.typename IS
62 |   'Name of the Type';
63 |
64 | COMMENT ON COLUMN article.startdate IS
65 |   'Date of Start';
66 |
67 | COMMENT ON COLUMN article.expiredate IS
68 |   'Date of Expire';
69 |
70 | COMMENT ON COLUMN article.colourinit IS
71 |   'Colour ID';
72 |
73 | COMMENT ON COLUMN article.colourname IS
74 |   'Name of the Colour';
75 |
76 | COMMENT ON COLUMN article.sex IS
77 |   'Sex';
78 |
79 | COMMENT ON COLUMN article.picture IS
80 |   'Picture';
81 |
82 | COMMENT ON COLUMN article.baseprice IS
83 |   'Base Price';
84 |
85 | COMMENT ON COLUMN article.saleprice IS
86 |   'Sale Price';
87 |
88 | COMMENT ON COLUMN article.notes IS
89 |   'Notes';
```

```

40 | /* Inventory */
41 | CREATE TABLE inventory (
42 |     articlecode      VARCHAR2(16) NOT NULL,
43 |     barcode          VARCHAR2(12) NOT NULL,
44 |     sizes            VARCHAR2(5) NOT NULL,
45 |     currentbaseprice NUMBER(12, 2),
46 |     currentsaleprice NUMBER(12, 2),
47 |     consignment      NUMBER(3, 0),
48 |     consignmentrp    NUMBER(12, 2),
49 |     qty               NUMBER(3, 0) NOT NULL,
50 |     status            CHAR(1) NOT NULL,
51 |     CONSTRAINT pk_inventory PRIMARY KEY ( articlecode,
52 |                                         barcode ),
53 |     CONSTRAINT fk_articleinventory FOREIGN KEY ( articlecode )
54 |         REFERENCES article ( articlecode ),
55 |     CONSTRAINT ck_status CHECK ( status IN ( '0', '1' ) )
56 | );
57 |
58 |
59 |
60 |
61 |
62 |
63 |
64 |
65 |
66 |
67 |
68 |
69 |
70 |
71 |
72 |
73 |
74 |
75 |
76 |
77 |
78 |
79 |
80 |
81 |
82 |
83 |
84 |
85 |
86 |
87 |
88 |
89 |
90 |
91 |
92 |
93 |
94 |
95 |
96 |
97 |
98 |
99 |
100 |
101 |
102 |
103 |
104 |
105 |
106 |
107 |
108 |
109 | COMMENT ON COLUMN inventory.articlecode IS
110 |     'Article Code';
111 |
112 | COMMENT ON COLUMN inventory.barcode IS
113 |     'Barcode';
114 |
115 | COMMENT ON COLUMN inventory.sizes IS
116 |     'Sizes';
117 |
118 | COMMENT ON COLUMN inventory.currentbaseprice IS
119 |     'Current Base Price';
120 |
121 | COMMENT ON COLUMN inventory.currentsaleprice IS
122 |     'Current Sale Price';
123 |
124 | COMMENT ON COLUMN inventory.consignment IS
125 |     'Consignment';
126 |
127 | COMMENT ON COLUMN inventory.consignmentrp IS
128 |     'Consignent Resale Price';
129 |
130 | COMMENT ON COLUMN inventory.qty IS
131 |     'Quantity';
132 |
133 | COMMENT ON COLUMN inventory.status IS
134 |     'Status';
135 |
136 | /* Cashier */
137 | CREATE TABLE cashier (
138 |     notrans        VARCHAR2(16) NOT NULL,
139 |     datetrans      DATE NOT NULL,
140 |     typetrans      VARCHAR2(10),
141 |     notes          VARCHAR2(50),
142 |     userid         VARCHAR2(20),
143 |     referencetrans VARCHAR2(20),
144 |     CONSTRAINT pk_cashier PRIMARY KEY ( notrans )
145 | );
146 |
147 | COMMENT ON COLUMN cashier.notrans IS
148 |     'Transaction Number';
149 |
150 | COMMENT ON COLUMN cashier.datetrans IS
151 |     'Transaction Date';
152 |
153 | COMMENT ON COLUMN cashier.typetrans IS
154 |     'Transaction Type';
155 |
156 | COMMENT ON COLUMN cashier.notes IS
157 |     'Notes';
158 |
159 | COMMENT ON COLUMN cashier.userid IS
160 |     'User ID';
161 |
162 | COMMENT ON COLUMN cashier.referencetrans IS
163 |     'Reference Transaction';

```

```

69 | /* CashierPayment */
70 | CREATE TABLE cashierpayment (
71 |     id          CHAR(5) NOT NULL,
72 |     notrans      VARCHAR2(16) NOT NULL,
73 |     paidtype     VARCHAR2(20) NOT NULL,
74 |     cardinit    CHAR(3),
75 |     cardname    VARCHAR2(20),
76 |     cardnumber   VARCHAR2(20),
77 |     totalpaid    NUMBER(12, 2),
78 |     machinename  CHAR(3),
79 |     companycharge NUMBER(12, 2),
80 |     customercharge NUMBER(12, 2),
81 |     reffno       VARCHAR2(10),
82 |     progressivedisc VARCHAR2(10) NOT NULL,
83 |     CONSTRAINT pk_cashierpayment PRIMARY KEY ( id ),
84 |     CONSTRAINT fk_cashiercashierpayment FOREIGN KEY ( notrans )
85 |         REFERENCES cashier ( notrans )
86 | );

```

COMMENT ON COLUMN cashierpayment.id IS  
 'cashier ID';

COMMENT ON COLUMN cashierpayment.notrans IS  
 'transaction ID';

COMMENT ON COLUMN cashierpayment.paidtype IS  
 'type of payment';

COMMENT ON COLUMN cashierpayment.cardinit IS  
 'card ID';

COMMENT ON COLUMN cashierpayment.cardname IS  
 'card name';

COMMENT ON COLUMN cashierpayment.cardnumber IS  
 'card number';

COMMENT ON COLUMN cashierpayment.totalpaid IS  
 'total amount paid';

COMMENT ON COLUMN cashierpayment.machinename IS  
 'name of the machine';

COMMENT ON COLUMN cashierpayment.companycharge IS  
 'charge to company';

COMMENT ON COLUMN cashierpayment.customercharge IS  
 'charge to customer';

COMMENT ON COLUMN cashierpayment.reffno IS  
 'reference number';

COMMENT ON COLUMN cashierpayment.progressivedisc IS  
 'progressive discount';

```

88 | /* CashierDetail */
89 | CREATE TABLE cashierdetail (
90 |     notrans      VARCHAR2(16) NOT NULL,
91 |     articlecode   VARCHAR2(20) NOT NULL,
92 |     barcode       VARCHAR2(12) NOT NULL,
93 |     sizes        VARCHAR2(5) NOT NULL,
94 |     qty          NUMBER(3, 0) NOT NULL,
95 |     baseprice    NUMBER(12, 2),
96 |     saleprice    NUMBER(12, 2),
97 |     discounttype CHAR(1),
98 |     discountpercent NUMBER(5, 2),
99 |     discountvalue NUMBER(12, 2),
100 |     disexpenses  CHAR(1),
101 |     cashierconsignment NUMBER(3, 0),
102 |     cashierconsignmentrp NUMBER(12, 2),
103 |     subtotal     NUMBER(12, 2),
104 |     payment      NUMBER(12, 2),
105 |     CONSTRAINT pk_cashierdetail PRIMARY KEY ( notrans,
106 |                                         barcode ),
107 |     CONSTRAINT fk_cashiercashierdetail FOREIGN KEY ( notrans )
108 |         REFERENCES cashier ( notrans ),
109 |     CONSTRAINT fk_inventorycashierdetail FOREIGN KEY ( barcode,
110 |                                         articlecode )
111 |         REFERENCES inventory ( barcode,
112 |                                         articlecode ),
113 |     CONSTRAINT ck_discounttype CHECK ( discounttype IN ( '0', '1', '2', '3' ) ),
114 |     CONSTRAINT ck_disexpenses CHECK ( disexpenses IN ( '0', '1' ) )
115 | );

```

```
COMMENT ON COLUMN cashierdetail.notrans IS
    'transaction ID';

COMMENT ON COLUMN cashierdetail.articlecode IS
    'article code';

COMMENT ON COLUMN cashierdetail.barcode IS
    'barcode';

COMMENT ON COLUMN cashierdetail.sizes IS
    'size of the article';

COMMENT ON COLUMN cashierdetail.qty IS
    'quantity';

COMMENT ON COLUMN cashierdetail.baseprice IS
    'base price';

COMMENT ON COLUMN cashierdetail.saleprice IS
    'sale price';

COMMENT ON COLUMN cashierdetail.discounttype IS
    'discount type';

COMMENT ON COLUMN cashierdetail.discountpercent IS
    'discount percentage';

COMMENT ON COLUMN cashierdetail.discountvalue IS
    'discountvalue';

COMMENT ON COLUMN cashierdetail.discxpenses IS
    'discount availability';

COMMENT ON COLUMN cashierdetail.cashierconsignment IS
    'consignment';

COMMENT ON COLUMN cashierdetail.cashierconsignmentrp IS
    'consignment resale price';

COMMENT ON COLUMN cashierdetail.subtotal IS
    'subtotal';

COMMENT ON COLUMN cashierdetail.payment IS
    'payment';
```

## 7. Number of rows in each table

```
4  /* Article */
5  SELECT
6      COUNT(*) AS num_of_rows
7  FROM
8      article;
```

Query Result x  
SQL | All Rows Fetched: 1 in 0.014 seconds

	NUM_OF_ROWS
1	12445

```
10 /* Inventory */
11 SELECT
12     COUNT(*) AS num_of_rows
13 FROM
14     inventory;
```

Query Result x  
SQL | All Rows Fetched: 1 in 0.027 seconds

	NUM_OF_ROWS
1	34010

```
16 /* Cashier */
17 SELECT
18     COUNT(*) AS num_of_rows
19 FROM
20     cashier;
```

Script Output x | Query Result x  
SQL | All Rows Fetched: 1 in 0.014 seconds

	NUM_OF_ROWS
1	8008

```
22 /* CashierPayment */
23 SELECT
24     COUNT(*) AS num_of_rows
25 FROM
26     cashierpayment;
```

Script Output x | Query Result x  
SQL | All Rows Fetched: 1 in 0.014 seconds

	NUM_OF_ROWS
1	8009

```
28 /* CashierDetail */
29 SELECT
30     COUNT(*) AS num_of_rows
31 FROM
32     cashierdetail;
```

Script Output x | Query Result x  
SQL | All Rows Fetched: 1 in 0.014 seconds

	NUM_OF_ROWS
1	11559

## 8. Descriptive Analysis

```

4 /* Article */
5 /* Number of items from different vendors*/
6 SELECT
7     vendorname,
8     COUNT(*) AS num_of_items
9 FROM
10    article
11 GROUP BY
12    vendorname
13 ORDER BY
14    num_of_items DESC;

```

Query Result | Fetched 50 rows in 0.016 seconds

VENDORNAME	NUM_OF_ITEMS
1 CATFLY	1223
2 AURIKICK	1146
3 AURIBUDDY	1018
4 VIRIDIPAL	727

```

4 /* Article */
5 /* Number of items from different vendors*/
6 SELECT
7     vendorname,
8     COUNT(*) AS num_of_items
9 FROM
10    article
11 GROUP BY
12    vendorname
13 ORDER BY
14    num_of_items ;

```

Query Result | Fetched 50 rows in 0.016 seconds

VENDORNAME	NUM_OF_ITEMS
1 STYLEQUEST	1
2 STYLEEDITION	1
3 QUIKSILVER	1
4 FRESHTRK	1

CATFLY provides the most items, and STYLEQUEST, STYLEEDITION and others provides the least items.

```

16 /* Average, min, max base price for items from different vendors*/
17 SELECT
18     vendorname,
19     to_char(AVG(baseprice),
20             '9999999999.99') AS average_baseprice,
21     MIN(baseprice)          AS min_baseprice,
22     MAX(baseprice)          AS max_baseprice
23 FROM
24    article
25 GROUP BY
26    vendorname
27 ORDER BY
28    max_baseprice DESC;

```

Query Result | Fetched 50 rows in 0.017 seconds

VENDORNAME	AVERAGE_BASEPRICE	MIN_BASEPRICE	MAX_BASEPRICE
1 VIRIDIPAL	396001.48	8000	3070000

```

16 /* Average, min, max base price for items from different vendors*/
17 SELECT
18     vendorname,
19     to_char(AVG(baseprice),
20             '9999999999.99') AS average_baseprice,
21     MIN(baseprice)          AS min_baseprice,
22     MAX(baseprice)          AS max_baseprice
23 FROM
24    article
25 GROUP BY
26    vendorname
27 ORDER BY
28    min_baseprice;

```

Query Result | Fetched 50 rows in 0.017 seconds

VENDORNAME	AVERAGE_BASEPRICE	MIN_BASEPRICE	MAX_BASEPRICE
1 AURIKICK	66242.64	6600	250000

```

16 /* Average, min, max base price for items from different vendors*/
17 SELECT
18     vendorname,
19     to_char(AVG(baseprice),
20             '9999999999.99') AS average_baseprice,
21     MIN(baseprice)          AS min_baseprice,
22     MAX(baseprice)          AS max_baseprice
23 FROM
24    article
25 GROUP BY
26    vendorname
27 ORDER BY
28    average_baseprice DESC;

```

Query Result | Fetched 50 rows in 0.018 seconds

VENDORNAME	AVERAGE_BASEPRICE	MIN_BASEPRICE	MAX_BASEPRICE
1 HIPJAUNT	1565761.90	1199200	2599000

VIRIDIPAL has the highest base price, and AURIKICK has the lowest base price. HIPJAUNT has the highest average base price.

```

32 /* Average, min, max sale price for items from different vendors*/
33 SELECT
34     vendorname,
35     to_char(AVG(saleprice),
36             '9999999999.99') AS average_saleprice,
37     MIN(saleprice)          AS min_saleprice,
38     MAX(saleprice)          AS max_saleprice
39 FROM
40    article
41 GROUP BY
42    vendorname
43 ORDER BY
44    max_saleprice DESC;

```

Query Result | Fetched 50 rows in 0.018 seconds

VENDORNAME	AVERAGE_SALEPRICE	MIN_SALEPRICE	MAX_SALEPRICE
1 VIRIDIPAL	589013.75	10000	8500000

```

32 /* Average, min, max sale price for items from different vendors*/
33 SELECT
34     vendorname,
35     to_char(AVG(saleprice),
36             '9999999999.99') AS average_saleprice,
37     MIN(saleprice)          AS min_saleprice,
38     MAX(saleprice)          AS max_saleprice
39 FROM
40    article
41 GROUP BY
42    vendorname
43 ORDER BY
44    min_saleprice;

```

Query Result | Fetched 50 rows in 0.011 seconds

VENDORNAME	AVERAGE_SALEPRICE	MIN_SALEPRICE	MAX_SALEPRICE
1 VIRIDIPAL	589013.75	10000	8500000

```

32 /* Average, min, max sale price for items from different vendors*/
33 SELECT
34     vendorname,
35     to_char(AVG(saleprice),
36             '9999999999.99') AS average_saleprice,
37     MIN(saleprice)          AS min_saleprice,
38     MAX(saleprice)          AS max_saleprice
39 FROM
40    article
41 GROUP BY
42    vendorname
43 ORDER BY
44    average_saleprice DESC;

```

Query Result | Fetched 50 rows in 0.017 seconds

VENDORNAME	AVERAGE_SALEPRICE	MIN_SALEPRICE	MAX_SALEPRICE
1 HIPJAUNT	1875190.48	1499000	2999000

VIRIDIPAL has both the highest sale price and the lowest sale price. HIPJAUNT has the highest average sale price.

```

48 /* Average, min, max duration for items from different vendors*/
49 SELECT
50     vendorname,
51     to_char( AVG(expiredate - startdate),
52             '999999.99') AS average_duration,
53     MIN(expiredate - startdate) AS min_duration,
54     MAX(expiredate - startdate) AS max_duration
55 FROM
56     article
57 GROUP BY
58     vendorname
59 ORDER BY
60     max_duration DESC;

```

Query Result : Fetched 50 rows in 0.016 seconds

VENDORNAME	AVERAGE_DURATION	MIN_DURATION	MAX_DURATION
1 FRESHTOUR	93.93	90	990

```

48 /* Average, min, max duration for items from different vendors*/
49 SELECT
50     vendorname,
51     to_char( AVG(expiredate - startdate),
52             '999999.99') AS average_duration,
53     MIN(expiredate - startdate) AS min_duration,
54     MAX(expiredate - startdate) AS max_duration
55 FROM
56     article
57 GROUP BY
58     vendorname
59 ORDER BY
60     min_duration;

```

Query Result : Fetched 50 rows in 0.016 seconds

VENDORNAME	AVERAGE_DURATION	MIN_DURATION	MAX_DURATION
1 VIRIDIPAL	98.96	1	900

```

48 /* Average, min, max duration for items from different vendors*/
49 SELECT
50     vendorname,
51     to_char( AVG(expiredate - startdate),
52             '999999.99') AS average_duration,
53     MIN(expiredate - startdate) AS min_duration,
54     MAX(expiredate - startdate) AS max_duration
55 FROM
56     article
57 GROUP BY
58     vendorname
59 ORDER BY
60     average_duration DESC;

```

Query Result : Fetched 50 rows in 0.017 seconds

VENDORNAME	AVERAGE_DURATION	MIN_DURATION	MAX_DURATION
1 VIRIDIMATE	122.00	122	122

FRESHTOUR has the longest consignment duration, and VIRIDIPAL has the shortest consignment duration. VIRIDIMATE has the highest average consignment duration.

```

64 /* Number of items for different categories*/
65 SELECT
66     categoryname,
67     COUNT(*) AS num_of_items
68 FROM
69     article
70 GROUP BY
71     categoryname
72 ORDER BY
73     num_of_items DESC;

```

Query Result : All Rows Fetched: 14 in 0.014 seconds

CATEGORYNAME	NUM_OF_ITEMS
1 T-SHIRT	4823
2 HAT	1930
3 SWEATER	1435
4 ACCESSORIES	1189
5 BAG	894
6 JACKET	740
7 PANTS	733
8 SHIRT	316
9 SHOES	199
10 WALLET	73
11 POLO SHIRT	54
12 SANDALS	43
13 OVERALL	10
14 BUNDLE	6

T-SHIRT has the largest number of items; BUNDLE has the least number of items.

```

75 /* Number of items for different types of item*/
76 SELECT
77   typename,
78   COUNT(*) AS num_of_items
79 FROM
80   article
81 GROUP BY
82   typename
83 ORDER BY
84   num_of_items DESC;

```

**Query Result**

All Rows Fetched: 44 in 0.014 seconds

TYPENAME	NUM_OF_ITEMS
CASUAL	5164

```

75 /* Number of items for different types of item*/
76 SELECT
77   typename,
78   COUNT(*) AS num_of_items
79 FROM
80   article
81 GROUP BY
82   typename
83 ORDER BY
84   num_of_items;

```

**Query Result**

All Rows Fetched: 44 in 0.016 seconds

TYPENAME	NUM_OF_ITEMS
GLOVES	1
ORGANIZER	1

CASUAL has the largest number of items, GLOVES and ORGANIZER have the least number of items.

```

86 /* Number of items for different colour of item*/
87 SELECT
88   colourname,
89   COUNT(*) AS num_of_items
90 FROM
91   article
92 GROUP BY
93   colourname
94 ORDER BY
95   num_of_items DESC;

```

**Query Result**

Fetched 50 rows in 0.017 seconds

COLOURNAME	NUM_OF_ITEMS
BLACK	4729

```

86 /* Number of items for different colour of item*/
87 SELECT
88   colourname,
89   COUNT(*) AS num_of_items
90 FROM
91   article
92 GROUP BY
93   colourname
94 ORDER BY
95   num_of_items;

```

**Query Result**

Fetched 50 rows in 0.016 seconds

COLOURNAME	NUM_OF_ITEMS
MISTY BLACK	1

Most items are in BLACK, only an item is in MISTY BLACK.

```

97 /* Number of items for different sex*/
98 SELECT
99   sex,
100  COUNT(*) AS num_of_items
101 FROM
102   article
103 GROUP BY
104   sex
105 ORDER BY
106   num_of_items DESC;

```

**Query Result**

All Rows Fetched: 4 in 0.009 seconds

SEX	NUM_OF_ITEMS
M	12066

Most items are for Male.

```

109 /* Number of inventory for each item*/
110 SELECT
111   i.articlecode,
112   articlename,
113   COUNT(*) AS num_of_inventory
114 FROM
115   inventory i
116   JOIN article a
117   ON i.articlecode = a.articlecode
118 GROUP BY
119   i.articlecode,
120   articlename
121 ORDER BY
122   num_of_inventory DESC;

```

**Query Result**

Fetched 50 rows in 0.035 seconds

ARTICLECODE	ARTICLENAMES	NUM_OF_INVENTORY
AR.MKV-PL035	CHELBY BLACK	10

```

109 /* Number of inventory for each item*/
110 SELECT
111   i.articlecode,
112   articlename,
113   COUNT(*) AS num_of_inventory
114 FROM
115   inventory i
116   JOIN article a
117   ON i.articlecode = a.articlecode
118 GROUP BY
119   i.articlecode,
120   articlename
121 ORDER BY
122   num_of_inventory;

```

**Query Result**

Fetched 50 rows in 0.026 seconds

ARTICLECODE	ARTICLENAMES	NUM_OF_INVENTORY
FRG-PI0001	UNSTRUCTURED - MAROON	1

CHELBY BLACK has the largest number of inventories; UNSTRUCTURED - MAROON has the least number of inventories.

```

124 /* inventory by consignment */
125 SELECT
126   consignment,
127   COUNT(*) AS num_of_inventory
128 FROM
129   inventory
130 GROUP BY
131   consignment
132 ORDER BY
133   num_of_inventory;

```

Query Result | SQL | All Rows Fetched: 6 in 0.009 seconds

CONSIGNMENT	NUM_OF_INVENTORY
1	10
2	15
3	20
4	25
5	30
6	0

CONSIGNMENT #0 has the largest number of inventories; CONSIGNMENT #10 has the least number of inventories.

```

135 /* consignment inventory by vendors */
136 SELECT
137   vendorname,
138   consignment,
139   COUNT(*) AS num_of_consignment_inventory
140 FROM
141   inventory i
142   JOIN article a
143     ON i.articlecode = a.articlecode
144 GROUP BY
145   vendorname,
146   consignment
147 ORDER BY
148   num_of_consignment_inventory DESC;

```

Query Result | SQL | Fetched 50 rows in 0.03 seconds

VENDORNAME	CONSIGNMENT	NUM_OF_CONSIGNMENT_INVENTORY
CATFLY	0	4086

```

136 SELECT
137   vendorname,
138   consignment,
139   COUNT(*) AS num_of_consignment_inventory
140 FROM
141   inventory i
142   JOIN article a
143     ON i.articlecode = a.articlecode
144 GROUP BY
145   vendorname,
146   consignment
147 ORDER BY
148   num_of_consignment_inventory;

```

Query Result | SQL | Fetched 50 rows in 0.014 seconds

VENDORNAME	CONSIGNMENT	NUM_OF_CONSIGNMENT_INVENTORY
STYLEQUEST	25	1

CATFLY has the largest number of consignment inventory, SATYLEQUEST has the least number of consignment inventory.

```

161 /* Cashier */
162 /* Frequency of employees handling payment*/
163 SELECT
164   userid,
165   COUNT(*) AS frequency
166 FROM
167   cashier
168 GROUP BY
169   userid
170 ORDER BY
171   frequency DESC;

```

Query Result | SQL | All Rows Fetched: 5 in 0.013 seconds

USERID	FREQUENCY
LISDA	3365

LISDA handles most of the payment.

```

183 /* Frequency of payment method */
184 SELECT
185   paidtype,
186   COUNT(*) AS frequency
187 FROM
188   cashierpayment
189 GROUP BY
190   paidtype
191 ORDER BY
192   frequency DESC;

```

Query Result | SQL | All Rows Fetched: 4 in 0.016 seconds

PAIDTYPE	FREQUENCY
CASH	7118

Most customers prefer to pay by cash.

```

194 /* Frequency of card */
195 SELECT
196     cardname,
197     COUNT(*) AS frequency
198 FROM
199     cashierpayment
200 WHERE
201     cardname != '-'
202 GROUP BY
203     cardname
204 ORDER BY
205     frequency DESC;

```

Query Result | SQL | All Rows Fetched: 10 in 0.013 seconds

CARDNAME	FREQUENCY
1 Debit CBA	468

Most customers prefer to pay by Debit CBA.

```

207 /* Frequency of Machine used*/
208 SELECT
209     machinename,
210     COUNT(*) AS frequency
211 FROM
212     cashierpayment
213 WHERE
214     machinename != '-'
215 GROUP BY
216     machinename
217 ORDER BY
218     frequency DESC;

```

Query Result | SQL | All Rows Fetched: 5 in 0.017 seconds

MACHINENAME	FREQUENCY
1 CBA	760

The company mostly use machine from CBA.

```

241 /* Average,total, min, max of totalpaid */
242 SELECT
243     to_char(AVG(totalpaid),
244     '9999999990.99') AS average_totalpaid,
245     to_char(SUM(totalpaid),
246     '9999999990.99') AS total_totalpaid,
247     to_char(MIN(totalpaid),
248     '9999999990.99') AS min_totalpaid,
249     to_char(MAX(totalpaid),
250     '9999999990.99') AS max_totalpaid
251 FROM
252     cashierpayment;

```

Query Result | SQL | All Rows Fetched: 1 in 0.007 seconds

AVERAGE_TOTALPAID	TOTAL_TOTALPAID	MIN_TOTALPAID	MAX_TOTALPAID
235329.76	1884756060.00	0.00	4401000.00

The max totalpaid is 4,401,000.00, the min totalpaid is 0.00, and the average totalpaid is 235,329.76.

```

257 /* Find the bestseller item*/
258 SELECT
259     cd.articlecode,
260     a.articlename,
261     COUNT(*) AS frequency
262 FROM
263     cashierdetail cd
264     JOIN article a
265     ON cd.articlecode = a.articlecode
266 GROUP BY
267     cd.articlecode,
268     a.articlename
269 ORDER BY
270     frequency DESC;

```

Query Result | SQL | Fetched 50 rows in 0.025 seconds

ARTICLECODE	ARTICLENAME	FREQUENCY
1 ESH.AR-SE0001	VETRANS OLIVE GREEN	60

The bestseller item is VETRANS OLIVE GREEN.

```

268 |     a.articlename
269 | ORDER BY
270 |     frequency DESC;
271 |
272 | /* Frequency of discount type*/
273 | SELECT
274 |     discounttype,
275 |     COUNT(*) AS frequency
276 | FROM
277 |     cashierdetail
278 | GROUP BY
279 |     discounttype
280 | ORDER BY
281 |     frequency DESC;

```

Query Result x

SQL All Rows Fetched: 4 in 0.008 seconds

DISCOUNTTYPE	FREQUENCY
1	2
2	8728

The mostly used discount type is DISCOUNTTYPE #2.

```

281 | /* Average, total, min, max discount value */
282 | SELECT
283 |     to_char(AVG(discountvalue),
284 |             '999999999.99') AS average_discountvalue,
285 |     to_char(SUM(discountvalue),
286 |             '999999999.99') AS total_discountvalue,
287 |     to_char(MIN(discountvalue),
288 |             '999999999.99') AS min_discountvalue,
289 |     to_char(MAX(discountvalue),
290 |             '999999999.99') AS max_discountvalue
291 |
292 | FROM
293 |     cashierdetail;

```

Query Result x

SQL All Rows Fetched: 1 in 0.012 seconds

AVERAGE_DISCOUNTVALUE	TOTAL_DISCOUNTVALUE	MIN_DISCOUNTVALUE	MAX_DISCOUNTVALUE
43261.27	500057040.00	0.00	324000.00

The max discount value is 324,000.00, the min discount value is 0.00, and the average discount value is 43,261.27.

```

302 | /* Frequency of discount availability*/
303 | SELECT
304 |     disexpenses,
305 |     COUNT(*) AS frequency
306 |
307 | FROM
308 |     cashierdetail
309 | GROUP BY
310 |     disexpenses
311 | ORDER BY
312 |     frequency DESC;

```

Query Result x

SQL All Rows Fetched: 2 in 0.014 seconds

DISEXPENSES	FREQUENCY
1	8740
0	2819

8740 items have discount, while 2819 items do not have discount.

```

322 | /* Average ,total,min,max of consignment resale price*/
323 | SELECT
324 |     to_char(AVG(cashierconsignmentrp),
325 |             '999999999.99') AS average_consignmentrp,
326 |     to_char(SUM(cashierconsignmentrp),
327 |             '999999999.99') AS total_consignmentrp,
328 |     to_char(MIN(cashierconsignmentrp),
329 |             '999999999.99') AS min_consignmentrp,
330 |     to_char(MAX(cashierconsignmentrp),
331 |             '999999999.99') AS max_consignmentrp
332 |
333 | FROM
334 |     cashierdetail;

```

Query Result x

SQL All Rows Fetched: 1 in 0.017 seconds

AVERAGE_CONSIGNMENTRP	TOTAL_CONSIGNMENTRP	MIN_CONSIGNMENTRP	MAX_CONSIGNMENTRP
36789.92	425254734.00	5250.00	187500.00

The max consignment resale price is 187,500.00; the min consignment resale price is 5,250.00, and the average consignment resale price is 36,789.92.

```

335 /* Average, total, min,max of subtotal */
336 SELECT
337   to_char(AVG(subtotal),
338   '9999999999.99') AS average_subtotal,
339   to_char(SUM(subtotal),
340   '9999999999.99') AS total_subtotal,
341   to_char(MIN(subtotal),
342   '9999999999.99') AS min_subtotal,
343   to_char(MAX(subtotal),
344   '9999999999.99') AS max_subtotal
345 FROM
346 cashierdetail;

```

Query Result

	AVERAGE_SUBTOTAL	TOTAL_SUBTOTAL	MIN_SUBTOTAL	MAX_SUBTOTAL
1	163055.29	1884756060.00	28000.00	1250000.00

The max subtotal is 1,250,000.00, the min subtotal is 20,000.00 and the average subtotal is 163055.29.

```

348 /* Average, total, min, max payment*/
349 SELECT
350   to_char(AVG(payment),
351   '9999999999.99') AS average_payment,
352   to_char(SUM(payment),
353   '9999999999.99') AS total_payment,
354   to_char(MIN(payment),
355   '9999999999.99') AS min_payment,
356   to_char(MAX(payment),
357   '9999999999.99') AS max_payment
358 FROM
359 cashierdetail;

```

Query Result

	AVERAGE_PAYMENT	TOTAL_PAYMENT	MIN_PAYMENT	MAX_PAYMENT
1	126261.93	1459461666.00	9750.00	1062500.00

The max payment is 1,062,500.00, the min payment is 9,750.00, and the average payment is 126,261.93.

Tables are created for calculating discount percentage and the quantity sold.

```

367 CREATE TABLE discountpercent
368   AS
369   (
370     SELECT
371       articlecode,
372       barcode,
373       notrans,
374       discountvalue,
375       saleprice
376     FROM
377       cashierdetail
378   );
379
380 ALTER TABLE discountpercent ADD percentage NUMBER(5, 2);
381
382 UPDATE discountpercent
383 SET
384   percentage = discountvalue / saleprice;
385
386 CREATE TABLE quantitysold
387   AS
388   (
389     SELECT
390       articlecode,
391       barcode,
392       percentage,
393       COUNT(notrans) AS quantity
394     FROM
395       discountpercent
396     GROUP BY
397       articlecode,
398       barcode,
399       percentage
400   );

```

402 /\* Slope and intercept \*/
403 SELECT
404 to\_char(slope, '90.9999') AS slope,
405 to\_char(max\_y\_bar - (slope \* max\_x\_bar), '90.9999') AS intercept
406 FROM
407 (
408 SELECT
409 SUM((percentage - x\_bar) \* (quantity - y\_bar)) / SUM((percentage - x\_bar) \* (
410 percentage - x\_bar)) AS slope,
411 MAX(x\_bar),
412 max\_x\_bar,
413 MAX(y\_bar),
414 max\_y\_bar
415 FROM
416 (
417 SELECT
418 AVG(percentage) AS x\_bar,
419 AVG(quantity) AS y\_bar
420 FROM
421 quantitysold
422 ) qs,
423 quantitysold
424 );

Script Output

Query Result

	SLOPE	INTERCEPT
1	0.1359	2.2691

Slope: 0.1359

Intercept: 2.2691

As slope is positive, discount percentage is positively related to quantity sold.