

TDS3551

Data Management

NAME	ID	Role
Teo Yong Zheng	1142700997	MapReduce & Querying
Wong Zi Xiang	1142701102	Querying & Bonus
Andy Yong Jun Jie	1142701565	Documentation & Bonus

Table of Contents

Introduction 1

Fact-Based Model..... 1

Data Ingestion.....2

Querying.....5

MapReduce.....10

Bonus 14

Introduction

In this assignment, we will be processing transactional data of a certain cinema in Malaysia. We must utilize tools and techniques we had learned in this course such as Hadoop, Hive, Pig, MapReduce and Fact-Based modeling.

Fact-Based Model

The properties of data are:

1. **Raw** - The rawer your data, the more information you can deduce from it, and the more questions you can ask of it.
2. **Immutable** - For immutability in Big Data, you don't update or delete data, you only add more. Thus, human-fault tolerance and simplicity is increased.
3. **Perpetual** – Immutable data results in data to be eternally true. The exceptions are garbage collection and regulation.

The core properties of fact-based model are *Atomic* and *Timestamped*. For fact-based data modeling, we have separated our dataset into 7 facts:

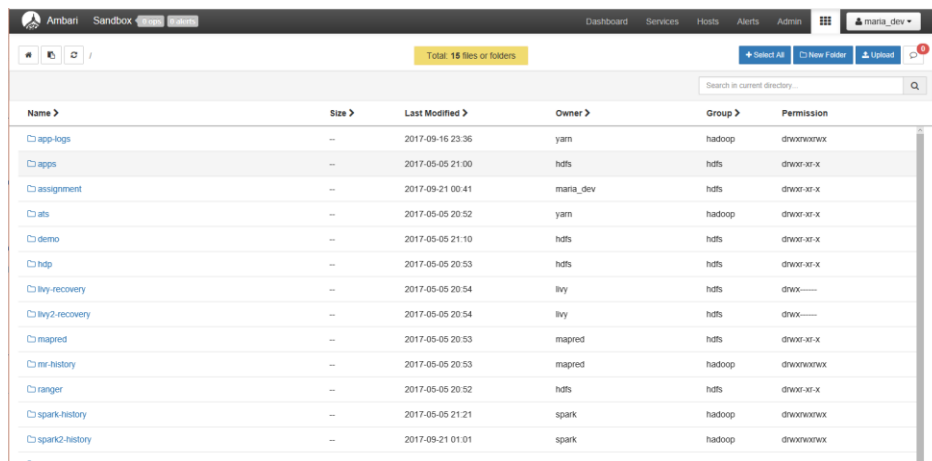
1. Transaction
2. Ticket
3. Film
4. Session
5. Sales
6. Screen
7. Workstation

Each fact contains, transaction number, sequence number, date time as compound key and variables categorized based on the fact:

1. Transaction – transaction type, transaction discount amount
2. Ticket – ticket type, ticket type code, ticket class, redemption, complimentary

3. Film – film, film code, film HO code
4. Session – session id, session screening time, session type, session status
5. Sales – sales tax, gross & net profit data
6. Screen – cinema name, screen name, seat number, screen row, screen layout
7. Workstation – user name, workstation name, order time, report code

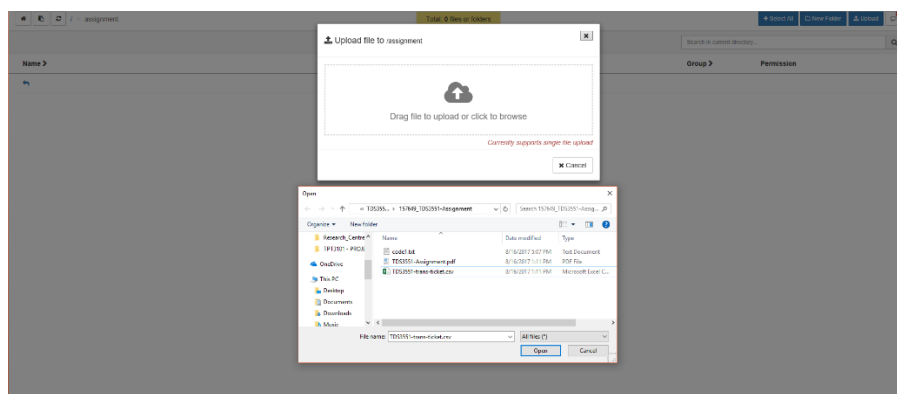
Data Ingestion



The screenshot shows the Ambari Sandbox interface with a file list. The top bar includes navigation links like Dashboard, Services, Hosts, Alerts, Admin, and a user profile for 'maria_dev'. A yellow banner indicates 'Total: 15 files or folders'. Below this is a search bar and a table of files and folders.

Name >	Size >	Last Modified >	Owner >	Group >	Permission
app-logs	--	2017-09-16 23:36	yarn	hadoop	drwxr-xr-x
apps	--	2017-05-05 21:00	hdfs	hdfs	drwxr-xr-x
assignment	--	2017-09-21 00:41	maria_dev	hdfs	drwxr-xr-x
ats	--	2017-05-05 20:52	yarn	hadoop	drwxr-xr-x
demo	--	2017-05-05 21:10	hdfs	hdfs	drwxr-xr-x
hdp	--	2017-05-05 20:53	hdfs	hdfs	drwxr-xr-x
ivy-recovery	--	2017-05-05 20:54	ivy	hdfs	drwxr-xr-x
ivy2-recovery	--	2017-05-05 20:54	ivy	hdfs	drwxr-xr-x
mapred	--	2017-05-05 20:53	mapred	hdfs	drwxr-xr-x
mr-history	--	2017-05-05 20:53	mapred	hadoop	drwxr-xr-x
ranger	--	2017-05-05 20:52	hdfs	hdfs	drwxr-xr-x
spark-history	--	2017-05-05 21:21	spark	hadoop	drwxr-xr-x
spark2-history	--	2017-09-21 01:01	spark	hadoop	drwxr-xr-x

In *Files View*, create a new folder.



Inside the newly create folder, press the *Upload* icon on the top right corner and choose *TDS3551-trans-ticket.csv* to upload.

Name	Size	Last Modified	Owner	Group	Permission
TDS3551-trans-ticket.csv	40.2 MB	2017-09-21 01:00	mapr_guy	hdfs	-rwxr-xr--

Dataset uploaded successfully.

create table cinema (col_value STRING);

LOAD DATA INPATH '/assignment/TDS3551-trans-ticket.csv' OVERWRITE INTO TABLE cinema;

CREATE TABLE cinema_transactions (insert overwrite table cinema_transactions
Transaction_Number INT,	SELECT
Transaction_Sequence_Number INT,	regexp_extract(col_value, '^(?:([^\,]*),?)\{1\}', 1) Transaction_Number ,
Transaction_Date_Time STRING,	regexp_extract(col_value, '^(?:([^\,]*),?)\{2\}', 1) Transaction_Sequence_Number ,
Type_Of_Transaction STRING,	regexp_extract(col_value, '^(?:([^\,]*),?)\{3\}', 1) Transaction_Date_Time ,
Cinema STRING,	regexp_extract(col_value, '^(?:([^\,]*),?)\{4\}', 1) Type_Of_Transaction ,
Session_Id INT,	regexp_extract(col_value, '^(?:([^\,]*),?)\{5\}', 1) Cinema ,
Session_Screening_Time STRING,	regexp_extract(col_value, '^(?:([^\,]*),?)\{6\}', 1) Session_Id ,
Session_Type STRING,	regexp_extract(col_value, '^(?:([^\,]*),?)\{7\}', 1) Session_Screening_Time ,
Public_Screening STRING,	regexp_extract(col_value, '^(?:([^\,]*),?)\{8\}', 1) Session_Type ,
Seat_Allocation STRING,	regexp_extract(col_value, '^(?:([^\,]*),?)\{9\}', 1) Public_Screening ,
Session_Status STRING,	regexp_extract(col_value, '^(?:([^\,]*),?)\{10\}', 1) Seat_Allocation ,
Screen_Name STRING,	regexp_extract(col_value, '^(?:([^\,]*),?)\{11\}', 1) Session_Status ,
Film STRING,	regexp_extract(col_value, '^(?:([^\,]*),?)\{12\}', 1) Screen_Name ,
Film_Code STRING,	regexp_extract(col_value, '^(?:([^\,]*),?)\{13\}', 1) Film ,
Film_HO_Code STRING,	regexp_extract(col_value, '^(?:([^\,]*),?)\{14\}', 1) Film_Code ,
	regexp_extract(col_value, '^(?:([^\,]*),?)\{15\}', 1) Film_HO_Code ,
Ticket_Type STRING,	regexp_extract(col_value, '^(?:([^\,]*),?)\{16\}', 1) Ticket_Type ,
Ticket_Type_Code STRING,	regexp_extract(col_value, '^(?:([^\,]*),?)\{17\}', 1) Ticket_Type_Code ,
Ticket_Type_HO_Code STRING,	regexp_extract(col_value, '^(?:([^\,]*),?)\{18\}', 1) Ticket_Type_HO_Code ,
Ticket_Class STRING,	regexp_extract(col_value, '^(?:([^\,]*),?)\{19\}', 1) Ticket_Class ,
	regexp_extract(col_value, '^(?:([^\,]*),?)\{20\}', 1) Complimentary ,
Complimentary STRING,	regexp_extract(col_value, '^(?:([^\,]*),?)\{21\}', 1) Redemption ,
Redemption STRING,	regexp_extract(col_value, '^(?:([^\,]*),?)\{22\}', 1) Price_Card ,
Price_Card STRING,	regexp_extract(col_value, '^(?:([^\,]*),?)\{23\}', 1) Sales_Tax ,
Sales_Tax STRING,	regexp_extract(col_value, '^(?:([^\,]*),?)\{24\}', 1) Admits ,
Admits INT,	regexp_extract(col_value, '^(?:([^\,]*),?)\{25\}', 1) Gross_Box_Office ,
Gross_Box_Office INT,	regexp_extract(col_value, '^(?:([^\,]*),?)\{26\}', 1) Net_BoxOffice ,
Net_BoxOffice INT,	regexp_extract(col_value, '^(?:([^\,]*),?)\{27\}', 1) Sales_Tax_1 ,
Sales_Tax_1 INT,	regexp_extract(col_value, '^(?:([^\,]*),?)\{28\}', 1) Sales_Tax_2 ,

Sales_Tax_2 INT,	regex_extract(col_value, '^?:([^,]*,)?{29}', 1) Gross_Box_Office_Each ,
Gross_Box_Office_Each INT,	regex_extract(col_value, '^?:([^,]*,)?{30}', 1) Sales_Tax_1_Each ,
Sales_Tax_1_Each INT,	regex_extract(col_value, '^?:([^,]*,)?{31}', 1) Sales_Tax_2_Each ,
Sales_Tax_2_Each INT,	regex_extract(col_value, '^?:([^,]*,)?{32}', 1) Gross_Each_excl_Redemp ,
Gross_Each_excl_Redemp INT,	regex_extract(col_value, '^?:([^,]*,)?{33}', 1) Sales_Tax_1_excl_Redemp ,
Sales_Tax_1_excl_Redemp INT,	regex_extract(col_value, '^?:([^,]*,)?{34}', 1) Sales_Tax_2_excl_Redemp ,
Sales_Tax_2_excl_Redemp INT,	regex_extract(col_value, '^?:([^,]*,)?{35}', 1) `User` ,
`User` STRING,	regex_extract(col_value, '^?:([^,]*,)?{36}', 1) Workstation ,
Workstation STRING,	regex_extract(col_value, '^?:([^,]*,)?{37}', 1) Workstation_Desc ,
Workstation_Desc STRING,	regex_extract(col_value, '^?:([^,]*,)?{38}', 1) Workgroup_Desc ,
Workgroup_Desc STRING,	regex_extract(col_value, '^?:([^,]*,)?{39}', 1) Report_Code ,
Report_Code STRING,	regex_extract(col_value, '^?:([^,]*,)?{40}', 1) Reason ,
Reason STRING,	regex_extract(col_value, '^?:([^,]*,)?{41}', 1) Seat_Number ,
Seat_Number STRING,	regex_extract(col_value, '^?:([^,]*,)?{42}', 1) Screen_Row_ID ,
Screen_Row_ID STRING,	regex_extract(col_value, '^?:([^,]*,)?{43}', 1) Grid_Col_Id ,
Grid_Col_Id STRING,	regex_extract(col_value, '^?:([^,]*,)?{44}', 1) Grid_Row_Id ,
Grid_Row_Id STRING,	regex_extract(col_value, '^?:([^,]*,)?{45}', 1) Area_Category ,
Area_Category STRING,	regex_extract(col_value, '^?:([^,]*,)?{46}', 1) Area ,
Area STRING,	regex_extract(col_value, '^?:([^,]*,)?{47}', 1) Ticket_Num_numeric ,
Ticket_Num_numeric STRING,	regex_extract(col_value, '^?:([^,]*,)?{48}', 1) Ticket_Num_alpha ,
Ticket_Num_alpha STRING,	regex_extract(col_value, '^?:([^,]*,)?{49}', 1) POS_Session_ID ,
POS_Session_ID STRING,	regex_extract(col_value, '^?:([^,]*,)?{50}', 1) Order_Time_Secs ,
Order_Time_Secs INT,	regex_extract(col_value, '^?:([^,]*,)?{51}', 1) Sequence_Refunded ,
Sequence_Refunded INT,	regex_extract(col_value, '^?:([^,]*,)?{52}', 1) Parent_Ticket_Type ,
Parent_Ticket_Type STRING,	regex_extract(col_value, '^?:([^,]*,)?{53}', 1) Package_Group_Num ,
Package_Group_Num STRING,	regex_extract(col_value, '^?:([^,]*,)?{54}', 1) Ticket_Numbering_Style ,
Ticket_Numbering_Style STRING,	regex_extract(col_value, '^?:([^,]*,)?{55}', 1) Transaction_Discount_Amount ,
Transaction_Discount_Amount INT,	regex_extract(col_value, '^?:([^,]*,)?{56}', 1) Discount_Report_Code ,
Discount_Report_Code STRING,	regex_extract(col_value, '^?:([^,]*,)?{57}', 1) Full_Price ,
Full_Price INT,	regex_extract(col_value, '^?:([^,]*,)?{58}', 1) Layout
Layout STRING	from cinema;
);	# Create cinema and cinema transaction
PAGE 1	PAGE2

Querying

```
SELECT substr(Session_Screening_Time, 0, 9) AS SCREENING_DATE
, COUNT(1) COUNT
FROM cinema_transactions
GROUP BY substr(Session_Screening_Time, 0, 9)
ORDER BY COUNT DESC
LIMIT 3;
```

screening_date	count
15/4/2017	9874
14/4/2017	9191
16/4/2017	7405

From the result, we can see that 14th to 16th of April have the highest ticket sales. We found out that it is because 14th is Good Friday. Following the weekend ticket sales will likely increase because Christian is more likely to celebrate on 14th and the following weekend.

```
SELECT a.SCREENING_DATE, a.NUMBER_OF_TICKET, COUNT(1) COUNT
FROM
(
SELECT Transaction_Number
, substr(Session_Screening_Time, 0, 9) AS SCREENING_DATE
, Layout
, MAX(Transaction_Sequence_Number) AS NUMBER_OF_TICKET
FROM cinema_transactions
```

```

GROUP BY Transaction_Number, substr(Session_Screening_Time, 0, 9), Layout

ORDER BY NUMBER_OF_TICKET DESC

)a

WHERE a.SCREENING_DATE IN (

SELECT x.SCREENING_DATE FROM

(

SELECT substr(Session_Screening_Time, 0, 9) AS SCREENING_DATE

, SUM(FULL_PRICE) AS TOTAL_PRICE

FROM cinema_transactions

GROUP BY substr(Session_Screening_Time, 0, 9)

ORDER BY TOTAL_PRICE DESC

LIMIT 3

)x

)

GROUP BY a.SCREENING_DATE, a.NUMBER_OF_TICKET

ORDER BY COUNT DESC

LIMIT 5;

```


a.screening_date	a.number_of_ticket	count
15/4/2017	2	2157
14/4/2017	2	1877
16/4/2017	2	1766
15/4/2017	3	484
14/4/2017	3	456

By querying further based on previous result, we can see that tickets are mostly sold in a pair. The result is hinting that people that watches movie on these 3 days are most likely couple.

```

SELECT CONCAT(Screen_Row_ID, Seat_Number) SEAT, Layout, COUNT(1) HEAD_COUNT
FROM cinema_transactions
WHERE Layout IN
( SELECT a.Layout FROM
  (
    SELECT Layout, COUNT(1) COUNT
    FROM cinema_transactions
    GROUP BY Layout
    ORDER BY COUNT DESC
    LIMIT 1
  )a
)
GROUP BY CONCAT(Screen_Row_ID, Seat_Number), Layout
ORDER BY HEAD_COUNT DESC

```

```
LIMIT 3;
```

seat	layout	head_count
D8	Originalv2	1103
D7	Originalv2	1055
D6	Originalv2	1021

Through the query, we can see that the most popular seat is D8, D7 and D6. We can guess why these 3 seats are so popular based on common sense. People will usually choose the seats in the center to have the most comfortable viewing angle.

In the next queries, we would like to find out the average order time of tickets of the popular seats compared to total average order time of tickets without popular seats.

```
SELECT AVG(Order_Time_Secs) Order_Time

FROM cinema_transactions

where Order_Time_Secs > 0

AND CONCAT(Screen_Row_ID, Seat_Number) NOT LIKE "NA"

AND CONCAT(CONCAT(Screen_Row_ID, Seat_Number), Layout) IN (

    SELECT CONCAT(b.SEAT, b.Layout) FROM

(

SELECT CONCAT(Screen_Row_ID, Seat_Number) SEAT, Layout, COUNT(1) HEAD_COUNT

FROM cinema_transactions

WHERE Layout IN

( SELECT a.Layout FROM

(
```

```

SELECT Layout, COUNT(1) COUNT

FROM cinema_transactions

GROUP BY Layout

ORDER BY COUNT DESC

LIMIT 1

)a

)

GROUP BY CONCAT(Screen_Row_ID, Seat_Number), Layout

ORDER BY HEAD_COUNT DESC

LIMIT 3

)b

);

```

order_time

57.31875224094657

Query shows that people take an average of 57.3s when ordering tickets with popular seats.

```

SELECT AVG(Order_Time_Secs) Order_Time

FROM cinema_transactions

where Order_Time_Secs > 0

AND CONCAT(Screen_Row_ID, Seat_Number) NOT IN ('D8','D7','D6', "NA")

ORDER BY Order_Time ASC

LIMIT 10;

```

order_time

63.107036921395675

Query shows that people take an average of 63.1s when ordering tickets that is not popular seats.

We can see that there is a 9% increase in ordering time.

MapReduce

```
cinema = LOAD '/assignment/TDS3551-trans-ticket.csv' USING PigStorage(',');
raw_cinema = FILTER cinema BY $0 > 1;
```

```
cinema_details = FOREACH raw_cinema GENERATE
```

```
$0 AS Transaction_Number,
```

```
$1 AS Transaction_Sequence_Number ,
```

```
$2 AS Transaction_Date_Time ,
```

```
$3 AS Type_Of_Transaction ,
```

```
$4 AS Cinema ,
```

```
$5 AS Session_Id ,
```

```
$6 AS Session_Screening_Time ,
```

```
$7 AS Session_Type,
```

```
$8 AS Public_Screening ,
```

```
$9 AS Seat_Allocation ,
```

```
$10 AS Session_Status ,
```

```
$11 AS Screen_Name,
```

```
$12 AS Film,
```

```
$13 AS Film_Code,
```

```
$14 AS Film_HO_Code,
```

```
$15 AS Ticket_Type,
```

```
$16 AS Ticket_Type_Code,
```

```
$17 AS Ticket_Type_HO_Code,
```

```
$18 AS Ticket_Class,
```

```
$19 AS Complimentary,
```

```
$20 AS Redemption,
```

```
$21 AS Price_Card,
```

```
$22 AS Sales_Tax,
```

```
$23 AS Admits,
```

```
$24 AS Gross_Box_Office,
```

```
$25 AS Net_BoxOffice,
```

```
$26 AS Sales_Tax_1,
```

```
$27 AS Sales_Tax_2,
```

```
$28 AS Gross_Box_Office_Each,
```

```
$29 AS Sales_Tax_1_Each,
```

```
$30 AS Sales_Tax_2_Each,
```

```

$31 AS Gross_Each_excl_Redemp,
$32 AS Sales_Tax_1_excl_Redemp,
$33 AS Sales_Tax_2_excl_Redemp,
$34 AS User,
$35 AS Workstation,
$36 AS Workstation_Desc,
$37 AS Workgroup_Desc,
$38 AS Report_Code,
$39 AS Reason,
$40 AS Seat_Number,
$41 AS Screen_Row_ID,
$42 AS Grid_Col_Id,
$43 AS Grid_Row_Id,
$44 AS Area_Category,
$45 AS Area,
$46 AS Ticket_Num_numeric,
$47 AS Ticket_Num_alpha,
$48 AS POS_Session_ID,
$49 AS Order_Time_Secs,
$50 AS Sequence_Refunded,
$51 AS Parent_Ticket_Type,
$52 AS Package_Group_Num,
$53 AS Ticket_Numbering_Style,
$54 AS Transaction_Discount_Amount,
$55 AS Discount_Report_Code,
$56 AS Full_Price,
$57 AS Layout;

```

```

cinema = LOAD '/assignment/TDS3551-trans-ticket.csv' USING PigStorage(',');

raw_cinema = FILTER cinema BY $0 > 1;

map_reduce_1 = FOREACH raw_cinema GENERATE

$12 AS FILM,

SUBSTRING($6, 0, 9) AS SCREENING_DATE,

(double)$56 AS TOTAL_PRICE;

group_FILM = GROUP map_reduce_1 BY (FILM, SCREENING_DATE);

```

```

map_reduce_1_stg1 = FOREACH group_FILM GENERATE SUM(map_reduce_1.TOTAL_PRICE) AS
TOTAL_EARNING, group;

results = RANK map_reduce_1_stg1 by TOTAL_EARNING DESC;

limit_results = LIMIT results 10;

DUMP limit_results;

```

```

(1,70018.5,(Film 5,15/4/2017))
(2,67606.5,(Film 5,14/4/2017))
(3,52333.5,(Film 5,16/4/2017))
(4,44789.5,(Film 5,13/4/2017))
(5,25198.0,(Film 5,22/4/2017))
(6,23363.0,(Film 5,12/4/2017))
(7,22634.0,(Film 3,30/4/2017))
(8,22611.0,(Film 5,17/4/2017))
(9,21853.0,(Film 5,23/4/2017))
(10,20547.0,(Film 35,7/4/2017 ))

```

The results are grouped as index, total sales, (film code, screening date). From the result we can see that Film 5 have the highest sales at 15th April. According to our research, we found that *The Fate of the Furious* 8, one of the most popular movie series in the world is first screened in Malaysia on 13th April. 13th April do not have the highest sales is most likely because it is Wednesday. People are less busy on 14th April, Friday, thus boosting the sales.

Malaysia / The Fate of the Furious / Release date

April 13, 2017

Malaysia

```

cinema = LOAD '/assignment/TDS3551-trans-ticket.csv' USING PigStorage(',');

```

```

raw_cinema = FILTER cinema BY $0 > 1;

map_reduce_2 = FOREACH raw_cinema GENERATE

$0 AS Transaction_Number,

$1 AS Transaction_Sequence_Number ,

$12 AS FILM,

SUBSTRING($6, 0, 9) AS SCREENING_DATE,

(double)$56 AS TOTAL_PRICE;

group_trans_num = GROUP map_reduce_2 BY Transaction_Number;

map_reduce_2_stg1 = FOREACH group_trans_num GENERATE
MAX(map_reduce_2.Transaction_Sequence_Number) AS Number_of_ticket, group;

group_data_sq = GROUP map_reduce_2_stg1 BY Number_of_ticket;

map_reduce_2_stg2 = FOREACH group_data_sq GENERATE COUNT(map_reduce_2_stg1.Number_of_ticket)
AS Count_of_ticket, group;

results = RANK map_reduce_2_stg2 by Count_of_ticket DESC;

limit_data_results = LIMIT results 10;

DUMP limit_data_results;

```

```

(1,29198,2.0)(2,9316,1.0)(3,5756,3.0)(4,3980,4.0)(5,1513,5.0)(6,852,6.0)(7,286,7.0)(8,235,8.0)(9,123,10.0)(10,77,9.0)

```

Results are grouped as index, number of transactions and number of tickets bought in a single transaction. We can see that tickets are mostly sold in a pair, then in single followed by in triple. This means that most of the time, movie are watched by couple.

Bonus

In this bonus section, we are trying to find out which cinema this data belongs to. While researching Early Bird ticket type, we found that mmCineplexes have Early Bird Ticket promotion before 2pm excluding Wednesday and Public Holidays.

The Early Bird gets the worm, or cheap movie ticket. Ever wondered what to do early in the day or no plans? Why not enjoy a movie. Starting from ONLY RM6, watching movies are now cheap, best part, blockbuster movies are included.

Terms and Conditions:

- Please refer below promotion valid day for each location

Location	Available on
eCurve, Damansara	Mondays to Fridays (EXCLUDING Wednesdays and Public Holidays)
City Square, Johor Bahru	Mondays to Thursdays (EXCLUDING Wednesdays and Public Holidays)

- Early Bird Promotion is applicable for movies before **2pm**.

From figure below, we can see that there is no Early Bird ticket after 2pm and the last date Early Bird ticket sold on was 6th April, Thursday. mmCineplexes have two different Terms and Conditions for Damansara and Johor Bahru's branches. Thus, we can be certain that this dataset belongs to mmCineplexes Johor Bahru.

6/4/2017 13:53	Early Bird MYR8
6/4/2017 13:53	Early Bird MYR8
9/4/2017 20:14	Early Bird MYR8
9/4/2017 20:14	Early Bird MYR8
10/4/2017 10:10	Early Bird MYR8
10/4/2017 10:10	Early Bird MYR8

We try to further confirm this our suspect, mmCineplexes Johor Bahru.

```
SELECT DISTINCT screen_name FROM  
  
cinema_transactions  
  
WHERE  
  
screen_row_id IN ("P")
```



AND

```
seat_number IN ("28");
```

screen_name

H4

We found that the largest seat number 28, and largest row P, only appears in Hall 4 of the cinema. We then proceed to search for Hall 4 layout in mmCineplexes Johor Bahru.







Cineplex: mmCineplexes City Square, JB

Hall: HALL 4

Movie Title: Kingsman: The Golden Circle

Movie Date and Time: Thu 21 Sep 6:45PM

SELECT SEATS

 Your seat  Available  Sofa  Sold

S C R E E N																										
3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	P
3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	N
3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	M
3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	L
3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	K
3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	J
3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	H
3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	G
3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	F
3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	E
3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	D
3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	C
3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	B
3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	A

Next

As expected, the finding shows that largest row P and largest seat number 28 in Hall 4. Our prediction that the dataset belongs to mmCineplexes Johor Bahru is verified.

We also find out that film is *The fate of furious 8* in the mapreduce.