“Fortune” team

# Google Apps Application

This is the report(explanation & description) of our project

Github link: https://github.com/sessiyeva131/dbms-project/tree/main


Atchibay Saya 190103410
Sessiyeva Tomiris 190103459
Daudekenkyzy Darish 190103317
Yershege Yerkenaz 190103406


02.05.2021

# Introduction

This is our project for DBMS part 2. Our team has chosen Google Play Store dataset and has made an application for it. In our App, user can see all the apps and its details from dataset, search them and manipulate with filters and categories. Also, there're functions as INSERT, UPDATE, DELETE which lets user to add new app, change it or even delete. Our Application was made via JavaFX and SQL Developer. And idea was created by some brains.

Let's pull every single given task apart and explore our app…

# Main Part

Task №1: **Choose a dataset.**

- We chose Google Play Store dataset. And the link for it is down below; https://www.kaggle.com/lava18/google-play-store-apps
- Dataset has numeric values in such columns as 'Reviews';
- Dataset has 8000+ values, but we took only 500+ for good speed of actions;
- Dataset has 13 columns by default;

Task №2: **Build an Oracle database**

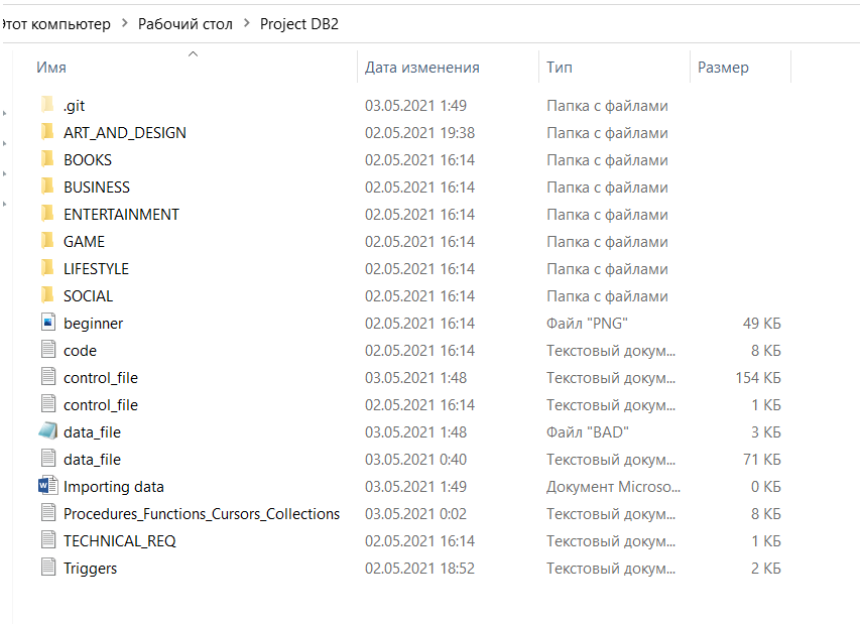
- We downloaded dataset as .txt file and loaded it to table 'app_dataset'.

```
create table app_dataset
(app_name VARCHAR2(400),
app_category VARCHAR2(30),
app_rating varchar2(30),
app_reviews number,
app_size varchar2(30),
installs VARCHAR2(30),
app_type VARCHAR2(30),
price VARCHAR2(20),
content_rating varchar2(30),
genres varchar2(70),
last_updated varchar2(50),
current_ver varchar2(30),
android_ver varchar2(30),
status varchar2(30) DEFAULT '-',
reminder varchar2(200) DEFAULT '-',
img BLOB
);
Table APP_DATASET created.
```

For inserting data we did next steps:

Here also you can check our code [https://github.com/sessiyeva131/dbms-project/blob/main/Importing_data.docx](https://github.com/sessiyeva131/dbms-project/blob/main/Importing_data.docx)

In our computer we have this folder where we hold all required data(data_file, control_file and images as well)




In data_file.txt we hold all data separated by comma:

```
WedMeGood - Wedding Planner,LIFESTYLE,4.6,1658,9.7M,"100,000+",Free,0,Everyone,Lifestyle,"July 31, 2018",2.0.9,4.4 and up
StyleSeat,LIFESTYLE,4.7,20304,24M,"500,000+",Free,0,Everyone,Lifestyle,"July 31, 2018",4.35,4.0.3 and up
DIY Garden Ideas,LIFESTYLE,4.1,3309,13M,"500,000+",Free,0,Everyone,Lifestyle,"August 5, 2018",10.0,4.0 and up
Brit + Co,LIFESTYLE,3.9,987,4.5M,"10,000+",Free,0,Everyone,Lifestyle,"August 29, 2017",2.0.4,4.0 and up
Creative Ideas - DIY & Craft,LIFESTYLE,4.0,5208,4.5M,"100,000+",Free,0,Teen,Lifestyle,"November 23, 2017",1.37,4.0 and up
Homestyler Interior Design & Decorating Ideas,LIFESTYLE,4.1,78298,29M,"5,000,000+",Free,0,Everyone,Lifestyle,"August 2, 2018",3.1.2.1,4.1 and up
JOANN - Crafts & Coupons,LIFESTYLE,4.6,34802,12M,"1,000,000+",Free,0,Everyone,Lifestyle,"June 22, 2018",5.2.2,4.3 and up
Wheretoget: Shop in style,LIFESTYLE,4.1,6808,12M,"500,000+",Free,0,Teen,Lifestyle,"June 5, 2017",3.2.0,4.1 and up
My Dressing - Fashion closet,LIFESTYLE,4.1,12452,Varies with device,"500,000+",Free,0,Everyone,Lifestyle,"January 26, 2017",Varies with
device,Varies with device
```

And we push all data which stores in data_file.txt into database by the usage of this control_file.txt.

```
LOAD DATA
INFILE 'data_file.txt'
   INTO TABLE APP_DATASET
   FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
   TRAILING NULLCOLS
        (
        APP_NAME,
        APP_CATEGORY,
        APP_RATING,
        APP_REVIEWS,
        APP_SIZE,
        INSTALLS,
        APP_TYPE,
        PRICE,
        CONTENT_RATING,
        GENRES,
        LAST_UPDATED,
        CURRENT_VER,
        ANDROID_VER,
        status,
        reminder,
        img_file_name filler char(100),
        img lobfile(img_file_name) TERMINATED by eof)
```

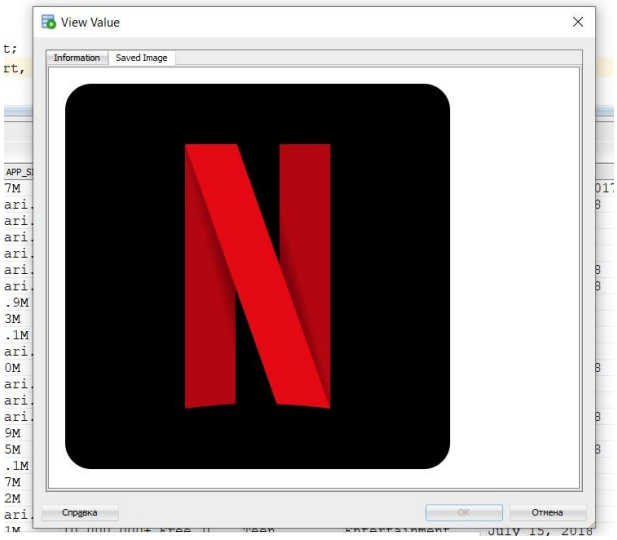
1) In command line we get into this folder and typed piece of code shown below:

```
C:\Users\User\Desktop\Project DB2>sqlldr system/plmoknqw@xe control=control_file.txt
```

Here system is my username of database and plmoknqw is the password. After typing this command all data appear in our database.

- Our column ‘IMG ’that holding images gets it as BLOB. And when we open it, it shows us our picture;

| | APP_NAME | APP_CATEGORY | AP... | APP_REVIEWS | APP_SIZE | INSTALLS | APP_... | PRICE | CONTENT_RATING | GENRES | LAST_UPDATED | CURRENT_VER | ANDROID_VER | IMG |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 524 | LINE Launcher | PERSONALI... | 4.4 | 733838 | 21M | 10,000,000+ | Free | 0 | Everyone | Personalization | June 29, 2018 | 2.4.25 | 4.0.3 and up | (BLOB |
| 525 | Asteroids 3D live wallpaper | PERSONALI... | 4.1 | 157495 | 6.4M | 10,000,000+ | Free | 0 | Everyone | Personalization | July 24, 2018 | 4.0.0.7 | 4.4 and up | (BLOB |
| 526 | Sun Rise Free Live Wallpaper | PERSONALI... | 4.3 | 864818 | .6M | 10,000,000+ | Free | 0 | Everyone | Personalization | January 31, 2018 | 4.8.3 | 4.0 and up | (BLOB |
| 527 | Wallpapers HD | PERSONALI... | 4.6 | 4849812 | .1M | 10,000,000+ | Free | 0 | Everyone | Personalization | November 28, ... | 1.7.1 | 3.0 and up | (BLOB |
| 528 | Tiger Live Wallpaper | PERSONALI... | 4.1 | 777247 | .1M | 5,000,000+ | Free | 0 | Teen | Personalization | May 31, 2018 | 2.7.2 | 4.0 and up | (BLOB |
| 529 | ZEDGEPIъЂhCh Ringtones & Wal... | PERSONALI... | 4.6 | 6466641 | Vari... | 100,000,... | Free | 0 | Teen | Personalization | July 19, 2018 | Varies wi... | Varies with de... | (BLOB |
| 530 | Backgrounds HD (Wallpapers) | PERSONALI... | 4.6 | 2390185 | Vari... | 100,000,... | Free | 0 | Teen | Personalization | August 4, 2018 | Varies wi... | Varies with de... | (BLOB |
| 531 | MORTAL KOMBAT X | GAME | 4.4 | 3039889 | 18M | 10,000,000+ | Free | 0 | Mature 17+ | Action | June 27, 2018 | 1.18.2 | 4.0 and up | (BLOB |
| 532 | Pixel Gun 3D: Survival shoot... | GAME | 4.5 | 448718 | 255M | 50,000,000+ | Free | 0 | Teen | Action | July 4, 2018 | 15.1.2 | 4.0.3 and up | (BLOB |
| 533 | Dropbox | PRODUCTIVITY | 4.4 | 1860844 | 61M | 500,000,... | Free | 0 | Everyone | Productivity | August 1, 2018 | Varies wi... | Varies with de... | (BLOB |
| 534 | KineMaster PIP,ъЂЊ Pro Video... | VIDEO PLA... | 4.5 | 1013867 | 32M | 50,000,000+ | Free | 0 | Everyone | Video Players ... | August 3, 2018 | 4.5.0.107... | 4.1 and up | (BLOB |
| 535 | Block City Wars + skins export | FAMILY | 4.5 | 76270 | 28M | 10,000,000+ | Free | 0 | Teen | Simulation | March 30, 2018 | 6.7.5 | 4.0 and up | (BLOB |
| 536 | Sniper 3D Gun Shooter: Free ... | GAME | 4.6 | 7657490 | Vari... | 100,000,... | Free | 0 | Mature 17+ | Action | August 2, 2018 | Varies wi... | Varies with de... | (BLOB |
| 537 | Bike Race Free - Top Motorcy... | GAME | 4.5 | 2586261 | Vari... | 100,000,... | Free | 0 | Everyone | Racing | July 31, 2018 | 7.7.9 | 4.2 and up | (BLOB |
| 538 | Dance School Stories - Dance... | FAMILY | 4.4 | 91171 | 36M | 1,000,000+ | Free | 0 | Everyone | Role Playing | August 3, 2018 | 1.0.8 | 4.1 and up | (BLOB |
| 539 | 3D Bowling | SPORTS | 4.1 | 107624 | 313M | 100,000,... | Free | 0 | Everyone | Sports | January 18, 2018 | 3.1 | 2.0.1 and up | (BLOB |
| 540 | Plants vs. Zombies FREE | FAMILY | 4.4 | 4064868 | 69M | 100,000,... | Free | 0 | Everyone 10+ | Strategy | July 6, 2018 | 2.2.00 | 4.1 and up | (BLOB |
| 541 | Dream League Soccer 2018 | SPORTS | 4.6 | 9873470 | 74M | 100,000,... | Free | 0 | Everyone | Sports | July 16, 2018 | 5.064 | 4.4 and up | (BLOB |
| 542 | Dumb Ways to Die 2: The Games | FAMILY | 4.2 | 1671658 | Vari... | 50,000,000+ | Free | 0 | Teen | Casual | July 12, 2018 | Varies wi... | Varies with de... | (BLOB |
| 543 | Shadow Fight 2 Special Edition | GAME | 4.5 | 10440 | 86M | 50,000+ | Paid | $... | Teen | Action | May 30, 2018 | 1.0.3 | 4.1 and up | (BLOB |
| 544 | VK | SOCIAL | 3.8 | 5793284 | Vari... | 100,000,... | Free | 0 | Mature 17+ | Social | August 3, 2018 | Varies wi... | Varies with de... | (BLOB |
| 545 | Clash of Kings : The King Of... | FAMILY | 4.2 | 2233681 | 97M | 50,000,000+ | Free | 0 | Teen | Strategy | August 1, 2018 | 3.44.0 | 2.3.3 and up | (BLOB |
| 546 | Google Sheets | PRODUCTIVITY | 4.3 | 496397 | Vari... | 100,000,... | Free | 0 | Everyone | Productivity | August 6, 2018 | Varies wi... | Varies with de... | (BLOB |
| 547 | Video Downloader - for Insta... | VIDEO PLA... | 4.8 | 332623 | 4.6M | 10,000,000+ | Free | 0 | Everyone | Video Players ... | July 27, 2018 | 1.1.58 | 4.1 and up | (BLOB |



- We created two derived columns that are: status and reminder.
  Status is made by installs amount and filled with GOLD, SILVER and BRONZE values automatically by trigger ‘status_trigg’. We think, user needs it, because they should see what is most installed right now and seeing just lots of numbers aren’t convenient.

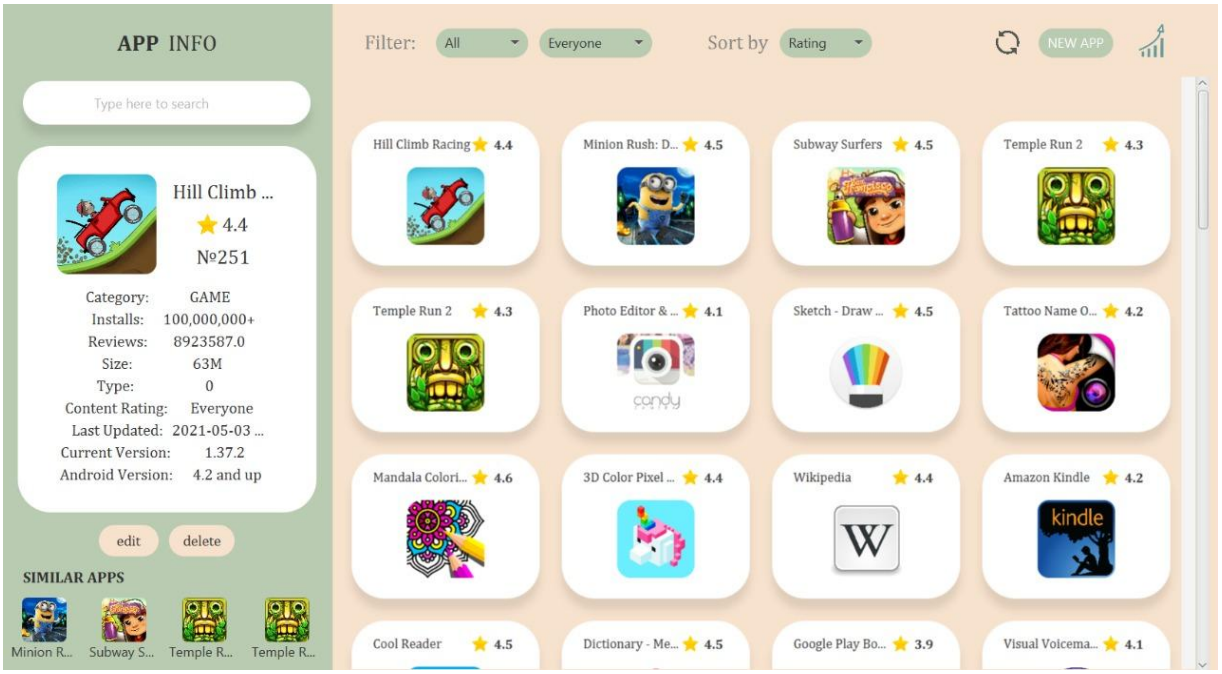
```
CREATE OR REPLACE TRIGGER status_trigg
BEFORE INSERT OR UPDATE OF installs ON app_dataset
FOR EACH ROW
BEGIN
    :new.status :=
    CASE
        WHEN num(:NEW.installs) >= 50000000
        THEN 'Gold'
        WHEN num(:NEW.installs) >= 10000000 AND num(:NEW.installs) < 50000000
        THEN 'Silver'
        WHEN num(:NEW.installs) >= 1000000 AND num(:NEW.installs) < 10000000
        THEN 'Bronze'
        ELSE '-'
    END;
END;
```

  Second derived column is reminder which is made by install number and filled automatically by trigger ‘REMINDER_TRIGG’. We need it, because we want to make interactions with developers too. And reminding them to improve the app or let know that the app is trending is important.

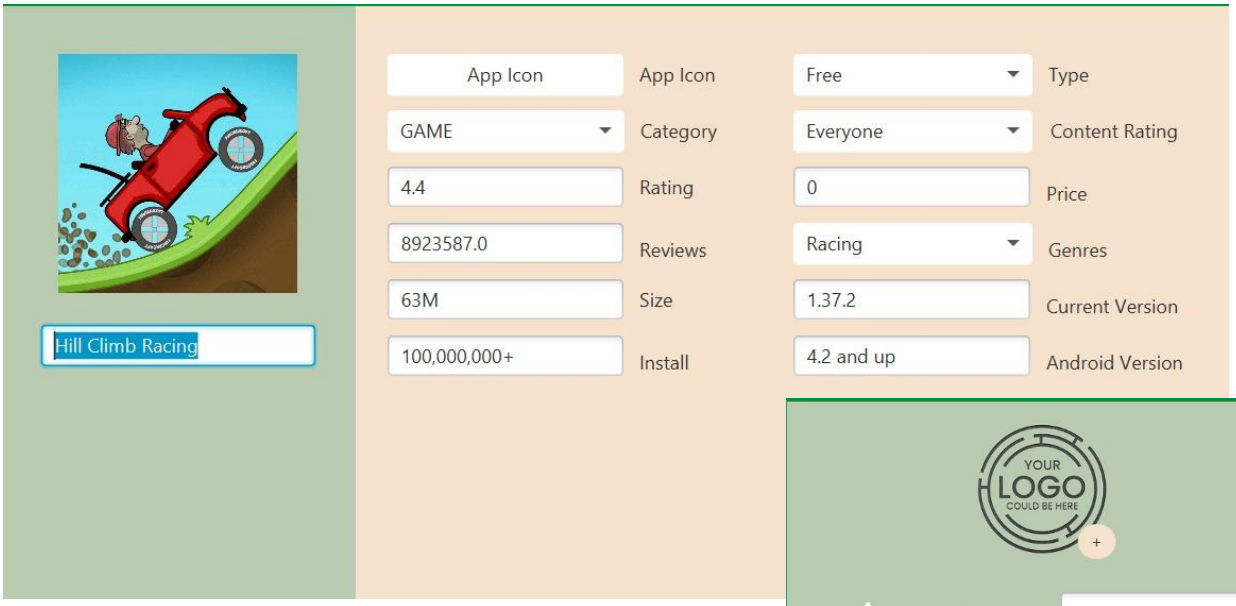
```
CREATE OR REPLACE TRIGGER REMINDER_TRIGG
BEFORE INSERT OR UPDATE OF APP_RATING, INSTALLS, LAST_UPDATED ON app_dataset
FOR EACH ROW
declare
    v_date varchar2(30);
BEGIN
    :new.REMINDER :=
    CASE
        WHEN
            num(:NEW.APP_RATING) < 40
            AND (num(:new.installs) <= 100000 AND num(:new.installs) > 10000)
        THEN 'DANGER! We suggested you to update your aplplication immediately!'
        WHEN
            num(:new.installs) <= 10000
        THEN 'WARNING! You should modify your application to increase users interest. Number of installs is too small!'
        WHEN
            num(:NEW.APP_RATING) > 46 AND num(:new.installs) >= 5000000
        THEN 'SUCCESS! Keep going! Your application is amazing! We are expecting even more apps from you.'
        ELSE '-'
    END;
END;
```

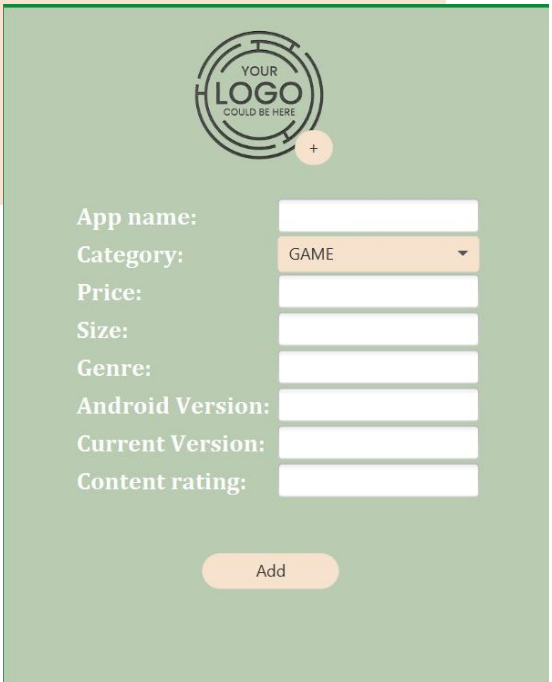
Task №3: **Create front part of your application**

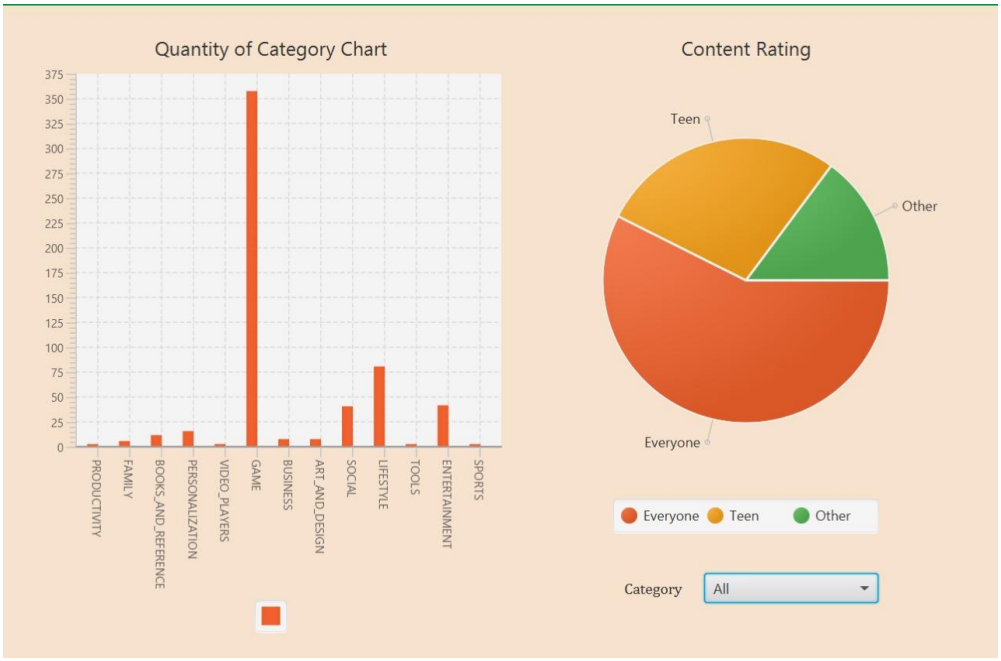
Let’s take a look at our application. Here you can see details of the App. Also, there are search (where you can find any app), filters and sorting to make user feel comfortable. And there you can see buttons as Insert, Update and Delete. Every icon is taken from dataset. Also you can see recommended apps below main (chosen) app.




And here’s Insert and Update windows.






And here's two statistics (Detailed explanation is in procedure part):

- Quantity of each Category;
- Content rating of each Category;

## Technical requirements

We completed all technical requirements, and let's take a look at them:

### Procedures:

```
1) PROCEDURES TO DEFINE TOP-CHART APPS
#---Main Procedure
create or replace procedure chart is
cursor cat_cur is                                   //CURSOR which holds app_categories
select distinct app_category
from app_dataset;
TYPE t_cat IS TABLE OF cat_cur%ROWTYPE              //INDEX BY TABLE, it holds app_categories with specific indexes
INDEX BY BINARY_INTEGER;
v_cat_tab t_cat;
v_count number := 1;
v_par VARCHAR2(1000);
plsql_block VARCHAR2(100);
begin
    plsql_block := 'BEGIN init(:a); END;';          //nested procedure
    for i in cat_cur loop
        v_cat_tab(v_count) := i;
        v_count := v_count + 1;
    end loop;
    for i in 1..v_cat_tab.COUNT loop
        v_par := v_cat_tab(i).app_category;
        EXECUTE IMMEDIATE plsql_block USING v_par;  //Usage of Dynamic SQL
    end loop;
end;
```

This procedure defines apps in top-chart. It's our main procedure that takes category name as parameter and passes it into init() to define each apps top chart place. Here we used cursor for categories 'cat_cur' and dynamic SQL 'plsql_block'.

```
#---Helper procedure
2)
create or replace procedure init(p_name varchar2) is
    cursor apps_cur is                                  //CURSOR, it holds applications ordered by their rating
    select app_name
    from app_dataset
    where app_category = p_name
    order by app_rating desc;
    type t_apps is table of apps_cur%ROWTYPE            //INDEX BY TABLE, for further usage of their indexes
    INDEX BY BINARY_INTEGER;
    v_apps_tab t_apps;
    v_count number := 1;
begin
    for i in apps_cur loop
        v_apps_tab(v_count) := i;
        v_count := v_count + 1;
    end loop;
    for i in 1..v_apps_tab.count loop
        update app_dataset
        set top_chart = i
        where app_name = v_apps_tab(i).app_name;
    end loop;
end;
```

Here in helper procedure we define top chart place of each app within given category.

==========

```
3)
#---Procedure to define top 5 used apps

create or replace procedure top(p_name varchar2) is
top_cur sys_refcursor;                                  //CURSOR, it holds applications ordered by their installs number
TYPE t_top IS TABLE OF app_dataset%ROWTYPE              //INDEX BY TABLE, for further usage of their indexes
INDEX BY BINARY_INTEGER;
v_top_tab t_top;
v_count number := 1;
l_row app_dataset%ROWTYPE;
begin
    OPEN top_cur FOR 'SELECT * FROM app_dataset WHERE                          //Dynamic SQL
         ' || p_name || ' order by installs desc';
    LOOP
    FETCH top_cur INTO l_row;
        v_top_tab(v_count) := l_row;
        v_count := v_count + 1;
        if v_count > 5
        then exit;
        end if;
    end loop;
    for i in 1..v_top_tab.COUNT loop
        update app_dataset
        set top_t = 3
        where app_name = v_top_tab(i).app_name;
    end loop;
end;

ALTER TABLE app_dataset ADD top_t number default 0;

begin
top('content_rating = ''Teen''');
end;
```

This procedure used in application as kind of “filter” that is defined by user to display top 5 apps. For instance, user can choose TOP 5 FREE Apps, Apps from any category or even content rating(Teen, Adult, etc).

# Function:

```
1)
##---Function which will convert chars to number

CREATE OR REPLACE FUNCTION num(str varchar2)
   RETURN number IS
   res NUMBER(20);
   str_res varchar2(30);
   BEGIN
      str_res := REPLACE(str, '+');
      str_res := REPLACE(str_res, ',');
      str_res := REPLACE(str_res, '"');
      res := TO_NUMBER(str_res, '9999999999');
      RETURN res;
END;
```

This function was needed while processing triggers and actions to compare installs amount between each other. So we converted chars into integers by this function.

==========

```
2)
##---Function which will return apps for recomendation system.

CREATE OR REPLACE PACKAGE rec_pkg AS                                    //Package, holds recommendation system's functions
    TYPE test_tab IS TABLE OF app_dataset%ROWTYPE;
    FUNCTION test_rec(l_where varchar2) RETURN test_tab PIPELINED;
END;

CREATE OR REPLACE PACKAGE BODY rec_pkg IS
FUNCTION test_rec(l_where VARCHAR2) RETURN test_tab PIPELINED IS
          cc sys_refcursor;
          l_row app_dataset%ROWTYPE;
       BEGIN
         OPEN cc FOR 'SELECT * FROM app_dataset WHERE
         app_category in (select app_category from app_dataset where ' || l_where|| ')';               //Dynamic SQL
          LOOP
             FETCH cc INTO l_row;
             EXIT WHEN cc%NOTFOUND;
            PIPE ROW (l_row);
         END LOOP;
         RETURN;
    END;
END;
```

In our application we have developed smart recommendation system. It always recommends by showing apps from same category.

==========

```
CREATE OR REPLACE PACKAGE charts_pkg AS                                  //Package, holds all statistic's functions
    function getValue(p_category varchar2) return number;
    function default_pieChart(p_name varchar2) return number;
    function pieChart(p_name varchar2, p_category varchar2) return number;
END;
CREATE OR REPLACE PACKAGE BODY charts_pkg IS                    //This is a package body

//-------------------------------------------------------------------------------------
//3)Function returns Y-value for corresponding X-value of BarChart

function getValue(p_category varchar2)
    return number is
        v_val number;
        begin
            select count(*) into v_val
            from app_dataset
            where app_category = p_category
            group by app_category;
            return v_val;
        end;
```

We have package where we hold three functions related to building of charts. First function. By that function get value we can get Y-axis' values for corresponding X-axis' values. By that we construct bar chart to represent "Quantity chart of each Category".

==========

```
//4)Function returns value for default pieChart using specified parameter

    function default_pieChart(p_name varchar2)
    return number is
    v_res number;
    v_count number;
    v_whole number;
    begin
        select count(app_name)
        into v_whole
        from app_dataset;
        if p_name = 'Everyone' then
            select count(*) into v_count
            from app_dataset
            where content_rating = 'Everyone'
            group by content_rating;
        elsif p_name = 'Teen' then
            select count(*) into v_count
            from app_dataset
            where content_rating = 'Teen'
            group by content_rating;
        else
            select sum(count(*)) into v_count
            from app_dataset
            where content_rating != 'Everyone' AND content_rating != 'Teen'
            group by content_rating;
        end if;
        v_res := round((v_count/v_whole) * 100);
        return v_res;
    end;
```

In this function we also get and define percentage rate for 'TEEN', 'EVERYONE' and 'Other' for our chart.

==========

```
    function pieChart(p_name varchar2, p_category varchar2)
    return number is
    v_res number;
    v_count number;
    v_whole number;
    begin
        select count(app_name)
        into v_whole
        from app_dataset
        where app_category = p_category;
        if p_name = 'Everyone' then
            select nvl(count(*), 0) into v_count
            from app_dataset
            where content_rating = 'Everyone' AND app_category = p_category
            group by content_rating;
        elsif p_name = 'Teen' then
            select nvl(count(*),0) into v_count
            from app_dataset
            where content_rating = 'Teen' AND app_category = p_category
            group by content_rating;
        else
            select nvl(sum(count(*)), 0) into v_count
            from app_dataset
            where content_rating != 'Everyone' AND content_rating != 'Teen' AND app_category = p_category
            group by content_rating;
        end if;
        v_res := round((v_count/v_whole) * 100);
        return v_res;
    end;
end;
```

This is more improved version of previous function. Here user can choose any category he likes to, and will get its content rating chart.

==========

**NOTE: We have used every single procedure and function in our application. And have presented you their usage.**

## Overall, we have:

```
FUNCTIONS(4) & PROCEDURES(3) -----> 8   //done
COLLECTIONS                  -----> 3   //done
CURSORS                      -----> 4   //done
PACKAGES                     -----> 2   //done
TRIGGERS                     -----> 3   //done
DYNAMIC SQL                  -----> 3   //done
```

## Conclusion.

This project was not easy one, but interesting and exiting one. We completed all database requirements and application part too. And we think, we put all our knowledge form first and second semester all together, to create this project. Thank you for attention and another good semester.

Atchibay Saya
Sessiyeva Tomiris
Daudekenkyzy Darish
Yershege Yerkenaz