CREATE table "REVIEWS" (

"ID" NUMBER GENERATED BY DEFAULT AS IDENTITY,

"USERNAME" VARCHAR2(60) NOT NULL,

"REVIEW" VARCHAR2(255),

"EVALUARE" VARCHAR2(2) NOT NULL,

constraint "REVIEWS\_PK" primary key ("ID")

)

/

CREATE sequence "REVIEWS\_SEQ"

/

CREATE trigger "BI\_REVIEWS"

before insert on "REVIEWS"

for each row

begin

if :NEW."ID" is null then

select "REVIEWS\_SEQ".nextval into :NEW."ID" from sys.dual;

end if;

end;

/

INSERT INTO REVIEWS VALUES ((SELECT COUNT(ID)+1 FROM REVIEWS), :V\_USERNAME, :V\_REVIEW, :V\_EVALUARE);

SELECT app, Last\_Updated

FROM Aplicatii

WHERE Last\_Updated > (SELECT Date from Aplicatii) ;

SELECT app, Last\_Updated

FROM Aplicatii

WHERE Last\_Updated = SELECT Last\_Updated from Aplicatii where date >= 12-05-2018;

select app, round(Rating,2) as Rating2018,

case

when round(Rating,2) > '4' then 'fa fa-arrow-up'

when round(Rating,2) >= '3' then 'fa fa-arrow-left'

else 'fa fa-arrow-down'

end Rating\_1,

1 as Rating\_2018

from Aplicatii

where Last\_updated >= '01.01.2018' AND Last\_Updated <= '12.31.2018';

<span class = "#RATING\_1#"></span>

CREATE OR REPLACE PROCEDURE afis\_id\_app (p\_id NUMBER) AS

v\_app aplicatii.app%TYPE;

v\_category aplicatii.category%TYPE;

v\_rating aplicatii.rating%TYPE;

v\_reviews aplicatii.reviews%TYPE;

v\_size aplicatii.size%TYPE;

v\_installs aplicatii.installs%TYPE;

v\_type aplicatii.type%TYPE;

BEGIN

SELECT app, category, rating, reviews, size, installs, type INTO v\_app, v\_category, v\_rating, v\_reviews, v\_size, v\_installs, v\_type

FROM APLICATII

WHERE ID = p\_id;

SYS.HTP.P ('Denumire aplicație: '||v\_project);

SYS.HTP.P ('Categorie: '||v\_task);

SYS.HTP.P ('Evaluari: '||v\_budget);

SYS.HTP.P ('Descărcări: '||v\_coord);

SYS.HTP.P ('Dimensiune '||v\_coord);

SYS.HTP.P ('Tipul: '||v\_coord);

SYS.HTP.P ('Dimensiune '||v\_coord);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('ID eronat !!!');

END;

/

Prod finala

CREATE OR REPLACE PROCEDURE afis\_id\_app (p\_id NUMBER) AS

v\_app aplicatii.app%TYPE;

v\_category aplicatii.category%TYPE;

v\_rating aplicatii.rating%TYPE;

v\_reviews aplicatii.reviews%TYPE;

v\_size\_ aplicatii.size\_%TYPE;

v\_installs aplicatii.installs%TYPE;

v\_type aplicatii.type%TYPE;

BEGIN

SELECT app, category, rating, reviews, size\_, installs, type INTO v\_app, v\_category, v\_rating, v\_reviews, v\_size\_, v\_installs, v\_type

FROM APLICATII

WHERE ID = p\_id;

SYS.HTP.P ('Denumire aplicație: '||v\_app||'<br>');

SYS.HTP.P ('Categorie: '||v\_category||'<br>');

SYS.HTP.P ('Rating: '||v\_rating||'<br>');

SYS.HTP.P ('Evaluari: '||v\_reviews||'<br>');

SYS.HTP.P ('Descărcări: '||v\_size\_||'<br>');

SYS.HTP.P ('Dimensiune '||v\_installs||'<br>');

SYS.HTP.P ('Tipul: '||v\_type||'<br>');

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE ('ID-ul este greșit! Vă rugăm să introduceți alt ID!');

END;

/

begin

htp.p('<table><tr><th>Luna</th><th>Numar Evaluari</th></tr>');

for x in (SELECT extract(MONTH FROM Last\_Updated) as Luna, SUM(reviews) as Numar\_Reviews

from Aplicatii

group by extract(month from Last\_Updated)

order by extract(month from Last\_Updated)) loop

htp.p('<tr>

<td>'||htf.escape\_sc(x.luna)||'</td>

<td>'||htf.escape\_sc(x.Numar\_Reviews)||'</td>

</tr>');

end loop;

htp.p('</table>');

end;