CS 327E Final Project

Natalie Gilliam, Jack Xiao, and Sean Moorhead
The Old News
The New Logical Data Model
create view FP_Vehicle_View as
SELECT  
    Character_id,  
    Character_name,  
    Created_Date,  
    Description,  
    Model,  
    Year,  
    Engine_Description
FROM  
    FP_Character
WHERE
    type = 'FP_Vehicle';

create or replace TRIGGER FP_Vehicle_trigger
INSTEAD OF insert ON FP_Vehicle_View
FOR EACH ROW
BEGIN
    insert into FP_Character(
        Character_id,  
        Character_name,  
        Created_Date,  
        Description,  
        Model,  
        Year,  
        Engine_Description,
        type)
    VALUES (  
        :New.Character_id,  
        :New.Character_name,  
        :New.Created_Date,  
        :New.Description,  
        :New.Model,  
        :New.Year,  
        :New.Engine_Description,  
        'FP_Vehicle');
END;
/
The DML!

--Disable foreign key constraints
alter table FP_PlayerPlatform disable constraint "FPR1" ;

--Truncate tables
truncate table FP_Platform;

-- insert into FP_Platform
INSERT INTO fp_platform ( platform_id, platform_name, version, release_date, supported) VALUES ( 1, 'NES', 1, TO_DATE('July 15, 1983', 'Month dd, YYYY'), 'N');

-- insert into FP_Game_Type
INSERT INTO fp_game_type ( game_type_id,game_type ) VALUES (1,'Racing' );

-- insert into FP_Soldier_View
INSERT INTO fp_soldier_view ( character_id, character_name, created_date, description, rank, specialty) VALUES (21,'Olivia Wileynose', TO_DATE('March 21, 2001', 'Month dd, YYYY'), 'A seasoned combat veteran ready for action.', 'Colonel', 'Explosives');

--Enable foreign key constraints
alter table FP_PlayerPlatform enable constraint "FPR1" ;
Our APEX Application

Ternary Relationship Select Lists:

- Name
- Game Title
- Character Name

We don’t want duplicate characters for a player and game to be added.
The Character select list:

```sql
select Character_name as d,
       Character_id as r
from FP_Character
WHERE Character_id NOT IN (SELECT Character_id FROM FP_Session WHERE :P26_Player_id = player_id and :P26_Game_id = game_id)
order by 1
```
Our Session Table only includes foreign keys, so we had to join with other tables to eliminate them. This SQL is shown below:

```sql
select
    session_id,
    name,
    Game_title,
    Character_name
from FP_Session S join FP_Character C on S.Character_id = C.Character_id
    join FP_Player P on S.Player_id = P.Player_id
    join FP_Game G on S.Game_id = G.Game_id
```
<table>
<thead>
<tr>
<th>Name</th>
<th>Game Title</th>
<th>Character Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vanessa Paz</td>
<td>Black Hawk Up</td>
<td>Pants on fire</td>
</tr>
<tr>
<td>Vanessa Paz</td>
<td>NBA 17</td>
<td>Pants on fire</td>
</tr>
<tr>
<td>Magda Remiszewski</td>
<td>HouseBuilder 7</td>
<td>So long, my friend</td>
</tr>
<tr>
<td>Jessica Bettaga</td>
<td>FIFA 17</td>
<td>Watch my butt</td>
</tr>
<tr>
<td>George Stathopoulos</td>
<td>FIFA 17</td>
<td>Watch my butt</td>
</tr>
<tr>
<td>Szyde Potter</td>
<td>APEX Developer 4</td>
<td>Universal No.1</td>
</tr>
<tr>
<td>Jack Xiao</td>
<td>Words with Acquaintances</td>
<td>Universal No.1</td>
</tr>
<tr>
<td>Magda Remiszewski</td>
<td>APEX Developer 4</td>
<td>Lilys Car</td>
</tr>
<tr>
<td>Jessica Bettaga</td>
<td>TonyHawk Gets Injured</td>
<td>Lilys Car</td>
</tr>
<tr>
<td>Magda Remiszewski</td>
<td>HouseBuilder 7</td>
<td>Lilys Car</td>
</tr>
<tr>
<td>Leslie Love</td>
<td>Super Smash Bros</td>
<td>Red Light for you</td>
</tr>
<tr>
<td>Millie Manson</td>
<td>LEGO 50 Shades of Grey</td>
<td>Green Light for me</td>
</tr>
<tr>
<td>Millie Manson</td>
<td>TonyHawk Gets Injured</td>
<td>CP3</td>
</tr>
</tbody>
</table>
Our APEX Application (cont.)

We also included a formatted classic report and formatted the columns with the following html:

```html
<span style="color:blue;font-size:150%;font-weight:bold">GAME_TYPE</span>

<\span style="color:green;font-size:350%;font-weight:bold">Number</\span>
```
<table>
<thead>
<tr>
<th>Game Type</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adventure</td>
<td>6</td>
</tr>
<tr>
<td>Combat</td>
<td>2</td>
</tr>
<tr>
<td>Sniper</td>
<td>3</td>
</tr>
<tr>
<td>Sports</td>
<td>7</td>
</tr>
<tr>
<td>Strategy</td>
<td>2</td>
</tr>
</tbody>
</table>
Our APEX Application (cont.)

We also wanted a chart with drill-down table. We made a pie chart and linked it to a table to show platform popularity.
To update our chart, we stored the selected Platform Name in a hidden page item. The report region used this hidden item in a WHERE clause. This SQL is shown here:

```
SELECT name as "Player Name",
    plat.platform_name || ' ' || plat.version AS "Platform and Version"
FROM FP_PlayerPlatform pp JOIN FP_Player p ON pp.player_id = p.player_id
    JOIN FP_Platform plat ON pp.platform_id = plat.platform_id
WHERE pp.platform_id IN (SELECT platform_id FROM FP_Platform WHERE platform_name = :P6_CHOSEN_PLATFORM)
```
Our APEX Application (cont.)

We also included a master-detail report. We linked a table of Characters to a table showing all the Players playing that Character.
This was done similarly to the chart drill-down, using a hidden page item to store the primary key of the selected Character. The SQL for the detail table is shown here:

```
SELECT name, game_title || ' ' || game_year as "Game"
FROM FP_Session s JOIN FP_Player p ON s.player_id = p.player_id
JOIN FP_Game g ON s.game_id = g.game_id
WHERE s.character_id = :P16_CHOSEN_CHARACTER
```
Our APEX Application (cont.)

Here is a link to our application for demo:

Thank you!