

CS 340: Project Grading Sheet Project

Group Number: 29

Group Members: Kaden Fugate, Luke Frugia, Samuel Rome, Derek Casini

URL <https://web.engr.oregonstate.edu/~fugateka/cs340/final/signup.php>

List below at least six features of your website. Give instructions on how to test each feature. All input must be error checked. Functionality implemented without SQL can be included for credit. However, you must have at least six SQL queries including at least one insert, update, delete generated from user input and at least two complex queries involving multiple tables.

1. SELECT 1 - To see your user's balance, create a user and sign in. After signing in, your balance will be displayed.

SQL:

```
SELECT user_name, balance
FROM USER
WHERE user_id = '$user_id'
```

2. SELECT 2 - To see upcoming games with odds, go to place a bet. You will see all of the upcoming games in different leagues with the odds of the home and away teams winning.

SQL:

```
SELECT G.game_id, G.game_date AS game_date, HT.city AS home_team, AT.city AS
away_team, O.home_win_odds, O.away_win_odds, L.name AS league_name
FROM GAME G
JOIN TEAM HT ON G.home_team_id = HT.team_id
JOIN TEAM AT ON G.away_team_id = AT.team_id
JOIN ODDS O ON G.game_id = O.odds_id
JOIN LEAGUE L ON HT.league_id = L.league_id
WHERE (G.game_date > CURDATE())
AND G.winning_team_id IS NULL)
```

3. SELECT 3 - A select is used to retrieve the bets that the user has made where the game has finished. To see this in action, you can see the 'past bets' table in user.php. This is done by joining the odds and game tables to get the bets where the game tables winning_team_id row is not null (aka a winner has been set).

SQL:

```
SELECT bet_id, bet_type, bet_date, bet_amount, payout, odds_id, G.winning_team_id,
home_team_id, away_team_id
FROM BETS_ON B
JOIN ODDS O ON B.odds_id = O.odds_id
JOIN GAME G ON O.odds_id = G.game_id
WHERE user_id = '{$user_id}'
AND G.winning_team_id IS NOT NULL
```

4. INSERT - When a bet is placed, the bet needs to be inserted into the BETS_ON table. The BETS_ON table holds all of the bets.

SQL:

```
INSERT INTO BETS_ON (bet_type, bet_date, bet_amount, payout, user_id, odds_id, game_id)
VALUES ('$bet_type', '$bet_date', '$bet_amount', '$payout', '$user_id', '$game_id', '$game_id')
```

5. UPDATE - An update is used to change the user's balance after choosing to double down on the user.php page. Because this is not an insert into the BETS_ON table, we also manually calculate the new payout by joining the ODDS table and doing some math.

SQL:

```
UPDATE BETS_ON B
JOIN ODDS O ON B.odds_id = O.odds_id
SET B.bet_amount = B.bet_amount * 2,
    B.payout = CASE B.bet_type
        WHEN 'home_win' THEN (B.bet_amount * 2) * O.home_win_odds
        WHEN 'away_win' THEN (B.bet_amount * 2) * O.away_win_odds
    END
WHERE bet_id = {$bet_id}
```

6. DELETE – A delete is used to delete existing bets. When a user deletes a bet, this is the SQL that is used.

SQL:

```
DELETE FROM BETS_ON
WHERE bet_id = {$bet_id};
```

7. Explain how to activate the trigger/procedure/function 1: A trigger is activated if a user places a bet. This is an AFTER INSERT trigger. The trigger updates the users balance by taking the placed amount out of the users account. To test this trigger, place a bet above 0.

8. Explain how to activate the trigger/procedure/function 2: A function is in place to calculate the payout of a bet. It does this by multiplying the odds by the bet amount. To test this, place any bet and the amount displayed within the payout column of the row corresponding to the bet that you made was calculated with this function.

9. Explain how to activate the trigger/procedure/function 3: A trigger is activated when the outcome of a game is posted. This is an AFTER UPDATE trigger. The trigger calls the payout_bets procedure on each bet placed on that game. To test this, place a winning bet on a game.