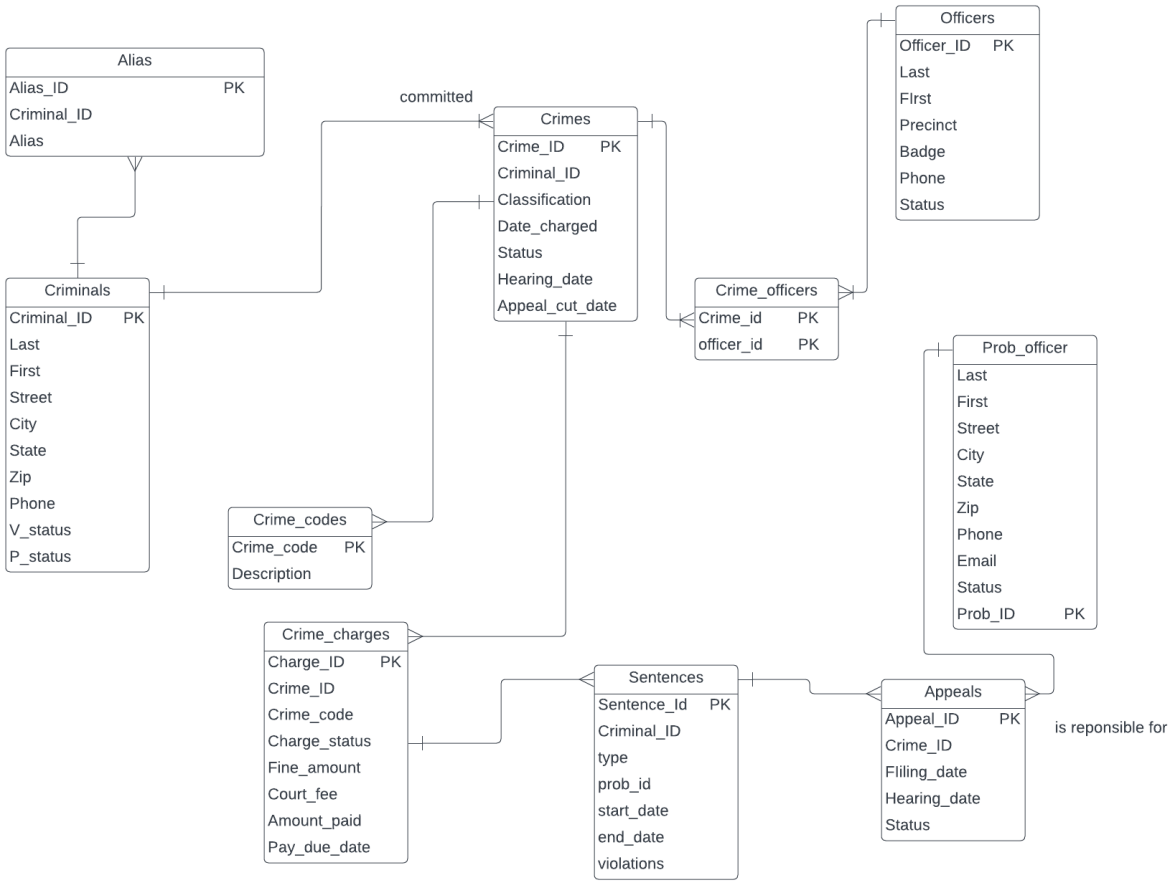
Chris Olmedo, David Chang
30 April 2023
Introduction to Databases

Milestone 4: Report

1) .



a)

b)

Alias(**Alias_ID**, Criminals.Criminal_ID, Alias)

Criminals(**Criminal_ID**, Last, First, Street, City, State, Zip, Phone, V_status, P_status)

Crimes(**crime_id**, Criminal.criminal_id, classification, date_charged, status, hearing_date, appeal_cut_date)

Sentences(**Sentence_ID**, Criminal.Criminal_ID, type, Prob_officers.Prob_ID, start_date, end_date, violations)

Prob_officer(**Prob_id**, Last, First, Street, City, State, Zip, Phone, Email, Status)

Crime_charges(**Charge_id,** Crimes.crime_id, Crime_codes.crime_code, charge_status, fine_amount, court_fee, amount_paid, pay_due_date)

Crime_officers(**Crimes.crime_id**, **Officers.officer_id**)

Officers(**officer_id**, last, first, precinct, badge, phone, status)

Appeals(**Appeal_ID**, Crimes.Crime_ID, filing_date, hearing_date, status)

Crime_codes(**crime_code**, description)

2) Database Programming
    a) The database is hosted locally using XAMPP. Access to the database as well as the local hosted webpages could be done through SSH key exchange. This allows anyone authorized such as superusers to access the webpage to change values. The local server doesn't need to be served on the internet for all to see as it poses an access control risk.
    b) The environment can be replicated by importing the database as well as the static php files needed.
    c) Install XAMPP on the local machine or instance. Run the SQL and Apache services. Clone the git repository to the htdocs directory under the XAMPP server. This should enable the php pages to be automatically located by the web server. Connect to localhost to test the connection.
    d) Advanced SQL procedures
        i) We have a procedure implemented to search Officers by their badge number. We utilized this procedure along with other similar procedures to search by any specific column that pertains to the associated record. For example, if we choose sort by precinct and enter a precinct number in the search bar, all officer records with that number are outputted. This does not change the database but changes the html of the page.
        ii) Another procedure implemented is the login function. We defined our login by a username and password in which it is validated against the database to see which permission type to assign to the session. This does not change the database but does change the user experience in that whatever webpages are available to the user, it would be determined by the permission_type.
3) Database Security
    a) We have two levels of database security, administrative access and user access.
    b) For developers, full root access is given to the database through the phpmyadmin interface. Since we isolated the database to a local machine, access control is instead implemented at this machine as opposed to the database itself. It is however protected by a username and password. A host user is given only read-only permissions.
    c) GRANT ALL PRIVILEGES ON *.* TO `root`@`localhost` WITH GRANT OPTION;
    GRANT PROXY ON ''@'%' TO 'root'@'localhost' WITH GRANT OPTION;

```
CREATE USER 'dev'@'%' IDENTIFIED VIA mysql_native_password
USING '***';GRANT ALL PRIVILEGES ON *.* TO 'dev'@'%' REQUIRE
NONE WITH GRANT OPTION MAX_QUERIES_PER_HOUR 0
MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0
MAX_USER_CONNECTIONS 0;GRANT ALL PRIVILEGES ON `jail`.* TO
'dev'@'%';
```

```
CREATE USER 'host'@'%' IDENTIFIED VIA mysql_native_password
USING '***';GRANT SELECT ON *.* TO 'host'@'%' REQUIRE NONE
WITH MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0
MAX_UPDATES_PER_HOUR 0 MAX_USER_CONNECTIONS 0;
```

4) Database Security (Application)
    a) We have implemented two types of accounts that have the ability to see certain pages, criminal and officer. The former is treated as a standard, read-only account that doesn't have access to changing values. The latter is somewhat a superuser, being able to edit account values and enter crime charges. Using the following code snippet, we could check if the session is logged in to an admin account or not.

```php
<?php
session_start();

// Check if the user is logged in
if (!isset($_SESSION["loggedin"]) ||
$_SESSION["loggedin"] !== true) {
    header("Location: login.php");
    exit;
}

// Check if the user is an admin
$is_admin = isset($_SESSION["is_admin"]) &&
$_SESSION["is_admin"] == "admin";
?>
```