Appendix

All Queries in Web:

1. Recommend TV shows based on ratings, director name, TvShow Year, or actor name based on user selection Recommend TV shows based on selected show's cast members AND genre

```
WITH Cast AS (
    SELECT CI.tid
    FROM Cast In CI
        JOIN IMDBPerson P
        ON CI.pid = P.pid
    WHERE P.name LIKE '%${actor}%'
), Genre AS (
    SELECT tid
    FROM Genres
    WHERE Genre LIKE '%${genre}%'
SELECT pTitle as TV_Show
FROM Titles
WHERE tid IN
    (SELECT tid FROM Cast)
    tid IN
    (SELECT tid FROM Genre)
ORDER BY TV_Show ASC
```

Display long running TV shows within specified rating range

```
SELECT t.pTitle AS TV_Show,
    (t.endYear - t.startYear)
    AS runTime
FROM Titles t JOIN Ratings r
    ON t.tid = r.tid
WHERE r.aveRating >=
    ${minInputRating}
    AND r.aveRating <=
    ${maxInputRating}
ORDER BY runTime DESC;</pre>
```

Display all directors' work who share the user's birth year

```
WITH TVDirectors AS (
    SELECT t.pTitle, d.pid
    FROM Directors d
        JOIN Titles t ON d.tid = t.tid
    WHERE t.type LIKE '%tvseries%'
)
SELECT td.pTitle AS TV_Show
FROM TVDirectors td
    JOIN IMDBPerson ip
    ON td.pid = ip.pid
WHERE ip.birthYear = ${user_birthyear};
```

Recommend TV shows based on ratings, director name, TvShow Year, or actor name based on user selection

```
SELECT DISTINCT t.tid AS tid,
    pTitle AS TV Show,
    aveRating AS Rating,
    startYear, endYear,
    "Add to WatchList" AS Watch
FROM Titles t
    JOIN Ratings r
    ON t.tid = r.tid
WHERE r.aveRating >= ${minRating}
    AND r.aveRating <= ${maxRating}</pre>
    AND t.startYear >= ${startYear}
    AND t.endYear <= ${endYear}</pre>
ORDER BY t.startYear, aveRating DESC;
WITH DirectorTitle AS (
    SELECT DISTINCT d.tid,
        ip.name AS Director
    FROM Directors d
        JOIN IMDBPerson ip
        ON d.pid = ip.pid
    WHERE ip.name LIKE
        "%${director_name}%"
),
    DirectorTVShows AS (
        SELECT d.tid,
            t.pTitle AS TV_Show,
            t.startYear,
            t.endYear, d.Director
        FROM DirectorTitle d
            JOIN Titles t
            ON d.tid = t.tid
    SELECT dtv.tid AS tid,
        TV Show, Director,
        aveRating AS Rating,
        startYear, endYear,
        "Add to WatchList" AS Watch
    FROM DirectorTVShows dtv
        JOIN Ratings r
        ON dtv.tid = r.tid
    WHERE r.aveRating >= ${minRating}
        AND r.aveRating <= ${maxRating}
        AND dtv.startYear >= ${startYear}
        AND dtv.endYear <= ${endYear}</pre>
    ORDER BY dtv.startYear,
        aveRating DESC,
        dtv.Director;
WITH ActorTitle AS (
    SELECT DISTINCT ci.tid,
        ip.name AS Actor
   FROM Cast In ci
```

```
JOIN IMDBPerson ip
                                                      FROM ActorTitle a
        ON ci.pid = ip.pid
                                                         JOIN Titles t
    WHERE ip.name LIKE "%${actor_name}%"
                                                          ON a.tid = t.tid
),
                                                  )
    ActorTVShows AS (
        SELECT a.tid,
                                        SELECT datv.tid AS tid,
            t.pTitle AS TV_Show,
                                             TV Show, Director,
            t.startYear,
                                             Actor, aveRating AS Rating,
                                             startYear, endYear,
            t.endYear, a.Actor
        FROM ActorTitle a
                                              "Add to WatchList" AS Watch
            JOIN Titles t
                                        FROM DirectorActorTVShows datv
                                             JOIN Ratings r
            ON a.tid = t.tid
                                             ON datv.tid = r.tid
    )
    SELECT atv.tid AS tid,
                                        WHERE r.aveRating >= ${minRating}
        TV_Show, Actor,
                                             AND r.aveRating <= ${maxRating}</pre>
        aveRating AS Rating,
                                             AND datv.startYear >= ${startYear}
                                             AND datv.endYear <= ${endYear}</pre>
        startYear, endYear,
        "Add to WatchList" AS Watch
                                        ORDER BY datv.startYear,
    FROM ActorTVShows atv
                                             aveRating DESC,
                                             datv.Director, datv.Actor;
        JOIN Ratings r
        ON atv.tid = r.tid
    WHERE r.aveRating >= ${minRating}2. Show based on range of runtimeMinutes; sorted
        AND atv.startYear >= ${startYear} SELECT DISTINCT t.tid,
        AND r.aveRating <= ${maxRating}</pre>
                                             t.pTitle as TV Show,
        AND atv.endYear <= ${endYear}</pre>
                                             t.runtimeMinutes AS Runtime,
    ORDER BY atv.startYear,
                                             "Add to WatchList" AS Watch
        aveRating DESC, atv.Actor;
                                         FROM Titles t JOIN Ratings r
                                             ON t.tid = r.tid
WITH DirectorTitle AS (
                                         WHERE t.runtimeMinutes
    SELECT DISTINCT d.tid,
                                             >=${minRangeRunTime}
        ip.name AS Director
                                             AND t.runtimeMinutes
    FROM Directors d
                                              <= ${maxRangeRunTime}
        JOIN IMDBPerson ip
                                         ORDER BY t.runtimeMinutes DESC;
        ON d.pid = ip.pid
    WHERE ip.name LIKE
                                     3. Display long running TV shows within specified rating
        "%${director name}%"
                                     range
),
                                         SELECT DISTINCT t.tid,
    ActorTitle AS (
        SELECT DISTINCT d.tid,
                                             t.pTitle AS TV_Show,
                                             (MAX(t.endYear)-MIN(t.startYear))
            d.Director,
                                             AS Years,
            ip.name AS Actor
                                             "Add to WatchList" AS Watch
        FROM DirectorTitle d
                                        FROM Titles t JOIN Ratings r
            JOIN Cast In ci
            ON d.tid = ci.tid
                                             ON t.tid = r.tid
            JOIN IMDBPerson ip
ON ci.pid = ip.pid
                                       GROUP BY t.tid
                                        HAVING Years>=${minYears}
                                             AND Years<=${maxYears}</pre>
        WHERE ip.name LIKE
                                ORDER BY Years DESC;
            "%${actor_name}%"
   ),
                                     4. Display all directors' work who share the user's birth
        DirectorActorTVShows AS (
                                     year
            SELECT a.tid,
                t.pTitle AS TV_Show,
                                         SELECT DISTINCT t.tid,
                Director, Actor,
                                            t.pTitle AS TV_Show,
                t.startYear, t.endYear
                                             IP.name as Director,
```

	"Add to WatchList" AS Watch	MTJ	H MIN AS (
	FROM Titles t JOIN Directors d		SELECT DISTINCT t.tid
	ON d.tid = t.tid		FROM Titles t
	JOIN IMDBPerson IP		WHERE t.type LIKE '%tvseries%'
	on IP.pid = d.pid		AND t.startYear<=\${birthYear}
	JOIN AdditionalTtles A	١	AND c.scarcical (V(Direnical)
)	ORI DOM M Lid Lid
	ON t.tid = A.tid		SELECT M.tid as tid,
	WHERE t.type LIKE '%tvseries%'		t.pTitle AS TV_Show,
	AND IP.birthYear=\${user_birthyear	:}	<pre>t.startYear, t.endYear,</pre>
	AND A.language="en";		"Add to WatchList" AS Watch
			FROM Titles t
	SELECT DISTINCT t.tid as tid,		JOIN MIN M
	t.pTitle AS TV_Show,		ON t.tid=M.tid
	IP.name as Director,		JOIN AdditionalTitles A
	"Add to WatchList" AS Watch		on t.tid = A.tid
	FROM Titles t		WHERE t.type LIKE '%tvseries%'
	JOIN Directors d		AND t.endYear>=\${birthYear}
	ON d.tid = t.tid		
			AND A.language="en";
	JOIN IMDBPerson IP		
	on IP.pid = d.pid	MTJ	'H MIN AS (
	WHERE t.type LIKE '%tvseries%'		SELECT DISTINCT t.tid
	AND IP.birthYear=\${user_birthyear	; };	FROM Titles t
			WHERE t.type LIKE '%tvseries%'
	Display all writers' work who share the user's birth		<pre>AND t.startYear<=\${birthYear}</pre>
year	•)	
			SELECT M.tid as tid,
	SELECT DISTINCT t.tid as tid,		t.pTitle AS TV_Show,
	t.pTitle AS TV_Show,		t.startYear, t.endYear,
	IP.name as Writer,		"Add to WatchList" AS Watch
	"Add to WatchList" AS Watch		FROM Titles t
	FROM Titles t		
	JOIN Writers w		JOIN MIN M
	ON w.tid = t.tid		ON t.tid=M.tid
	JOIN IMDBPerson IP		WHERE t.type LIKE '%tvseries%'
	on IP.pid = w.pid		<pre>AND t.endYear>=\${birthYear};</pre>
		7 Diamles	
			all shows within the range of the user's birth
		year	
	WHERE t.type LIKE '%tvseries%'	. CEI	ECT DISTINCT + +; d
	AND IP.birthYear=\${user_birthyear	;} SEI	ECT DISTINCT t.tid,
	AND A.language="en";		t.pTitle AS TV_Show,
			IP.name as Actor,
	SELECT DISTINCT t.tid as tid,		"Add to WatchList" AS Watch
	t.pTitle AS TV_Show,	FRC	M Titles t JOIN Cast_In CI
	<pre>IP.name as Writer,</pre>		ON CI.tid = t.tid
	"Add to WatchList" AS Watch		JOIN IMDBPerson IP
	FROM Titles t		ON IP.pid = CI.pid
	JOIN Writers w		JOIN AdditionalTitles A
	ON w.tid = t.tid		on t.tid = A.tid
		WHF	RE t.type LIKE '%tvseries%'
	JOIN IMDBPerson IP	*****	AND IP.birthYear=\${user_birthyear}
	on IP.pid = w.pid		-
	WHERE t.type LIKE '%tvseries%'		AND A.language="en";
	AND IP.birthYear=\${user_birthyear		DOM DIGHTNOM L L' 1
		SEI	ECT DISTINCT t.tid as tid,
b. I	Display all actors' shows who share the user's birth		t.pTitle AS TV_Show,
year			IP.name as Actor,

```
"Add to WatchList" AS Watch
                                                WITH DirectorList AS (
    FROM Titles t
                                                    SELECT pid, COUNT (DISTINCT tid)
        JOIN Cast In CI
                                                    AS numShows
        ON CI.tid = t.tid
                                                    FROM Directors
        JOIN IMDBPerson IP
                                                    GROUP BY pid
        on IP.pid = CI.pid
                                                    HAVING numShows >= ${minSize}
    WHERE t.type LIKE '%tvseries%'
                                                  ), DirectorsName AS (
                                                      SELECT IP.name, DL.numShows, D.tid
        AND IP.birthYear=${user_birthyear};
                                                      FROM IMDBPerson IP JOIN
8. Display TV shows and writers of a particular genre
                                                        DirectorList DL ON IP.pid = DL.pid
                                                        JOIN Directors D on IP.pid = D.pid
    WITH TVTitles AS (
        SELECT tid, pTitle
                                                    SELECT DISTINCT T.tid,
        FROM Titles
                                                        DN.name AS Director,
        WHERE type LIKE '%tvseries%'
                                                        T.pTitle as TV_Show, DN.numShows,
    ), GenresList AS (
                                                        "Add to WatchList" AS Watch
        SELECT T.tid
                                                    FROM Titles T JOIN DirectorsName DN
        FROM Titles T JOIN Genres G
                                                        ON T.tid=DN.tid
            ON T.tid = G.tid
                                                    WHERE DN.name LIKE "%${director}%"
        WHERE type LIKE '%tvseries%'
                                                    ORDER BY numShows DESC , name ASC;
            AND G.Genre LIKE '%${genre}%'
                                           11. Display all TV shows
    ), WritersList AS (
        SELECT w.pid, t.tid, t.pTitle
                                                SELECT DISTINCT T.tid as tid,
        FROM TVTitles t JOIN Writers w
                                                    T.pTitle as TV_Show,
            ON t.tid = w.tid
                                                    "Add to WatchList" AS Watch
        WHERE t.tid IN
                                                FROM Titles T
             (SELECT tid FROM GenresList)
                                                ORDER BY TV_Show ASC;
        GROUP BY w.pid
                                           12. Find all highest rated TV shows
    SELECT DISTINCT WL.tid,
        WL.pTitle as TV_Show,
                                                SELECT DISTINCT T.tid, T.pTitle as TV Show,
        IP.name as Writer
                                                    R.aveRating as Rating,
    FROM WritersList WL JOIN IMDBPerson IP
                                                    "Add to WatchList" AS Watch
        ON WL.pid = IP.pid
                                                FROM Titles T join Ratings R
    ORDER BY TV_Show;
                                                    on T.tid = R.tid
                                                WHERE type LIKE '%tvseries%'
9. Display TV shows that have cast sizes of at least min-
                                                    AND R.aveRating>=${minRating}
Size
                                                    AND R.aveRating <=${maxRating}</pre>
                                                ORDER BY Rating DESC;
    WITH SizeCount AS (
        SELECT tid , COUNT(*) AS size
                                           13. Populate Watch List
        FROM Cast In
                                                SELECT DISTINCT T.tid as tid,
        GROUP BY tid
                                                    pTitle AS TV Show,
                                                    "Remove from WatchList" AS Remove
    SELECT T.tid, T.pTitle as TV_Show,
                                                FROM Titles T
        SC.size
                                                    JOIN WatchList 1
    FROM SizeCount SC JOIN
                                                    ON T.tid = 1.tid
        Titles T ON SC.tid = T.tid
                                                ORDER BY pTitle;
    WHERE SC.size >= ${minSize}
        AND SC.size <=${maxSize}</pre>
                                           14. Add to Watch List
    ORDER BY SC.size DESC;
                                                INSERT INTO WatchList(tid)
10. Get tv show directed by a director involved in at
                                                VALUES ("${req.query.tid}");
least x-number (user input) of different TV shows within
a specified rating range
                                           15. Remove show from Watch List
```

```
DELETE FROM WatchList
WHERE tid = "${req.query.tid}";
```

16. Clear Watch List

DELETE FROM WatchList