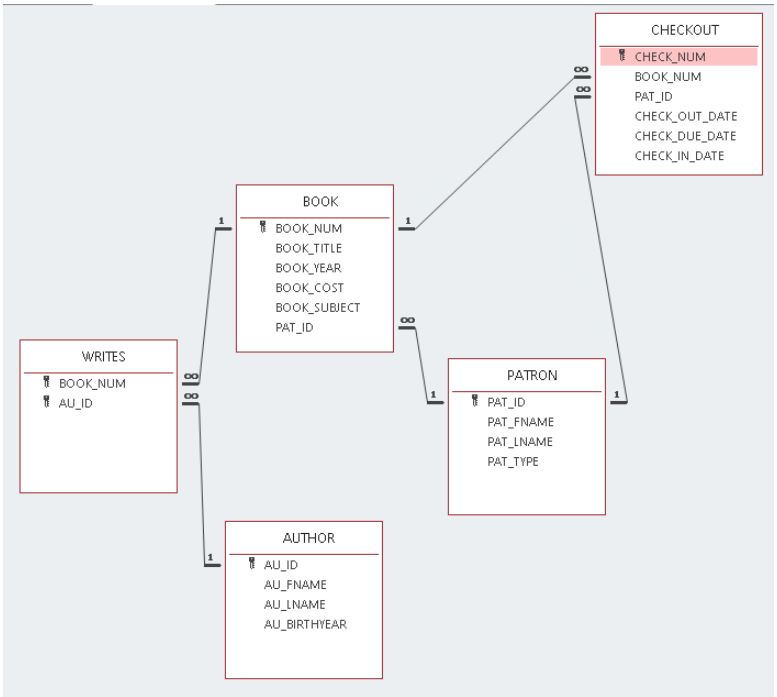
Team: Daddy's Darlings **Final Group Project** **Part 1** 11/21/2019

Team Members : Hasan Mohammad, Matthew Vetter, Matt Fields, Chase Sellers, David Karem, Nathan Brewer, Elijah Barnard, Jarod Dennis



56.
```
SELECT BOOK_TITLE, BOOK_COST, BOOK_YEAR
FROM BOOK
ORDER BY BOOK_TITLE;
```

57.
```
SELECT PAT_FNAME, PAT_LNAME
FROM PATRON
ORDER BY UPPER(PAT_LNAME), UPPER(PAT_FNAME);
```

58.
```
SELECT CHECK_NUM, CHECK_OUT_DATE, CHECK_DUE_DATE
FROM CHECKOUT
ORDER BY CHECK_NUM;
```

59.
```
SELECT BOOK_NUM, BOOK_TITLE AS TITLE, BOOK_SUBJECT AS "Subject of Book"
FROM BOOK
ORDER BY BOOK_NUM;
```

60.
```
SELECT DISTINCT BOOK_YEAR
FROM BOOK
ORDER BY BOOK_YEAR;
```

61.
```
SELECT DISTINCT BOOK_SUBJECT
FROM BOOK
ORDER BY BOOK_SUBJECT;
```

62.
```
SELECT BOOK_NUM, BOOK_TITLE, BOOK_COST AS "Replacement Cost"
FROM BOOK
ORDER BY BOOK_NUM;
```

63.
```
SELECT CHECK_NUM, BOOK_NUM, PAT_ID, CHECK_OUT_DATE, CHECK_DUE_DATE
FROM CHECKOUT
ORDER BY CHECK_OUT_DATE DESC, CHECK_NUM;
```

64.
```
SELECT BOOK_TITLE, BOOK_YEAR, BOOK_SUBJECT
FROM BOOK
ORDER BY BOOK_SUBJECT, BOOK_YEAR DESC, BOOK_TITLE;
```

65.
```
SELECT BOOK_NUM, BOOK_TITLE, BOOK_COST
FROM BOOK
WHERE BOOK_COST = 59.95
ORDER BY BOOK_NUM;
```

66.
```
SELECT BOOK_NUM, BOOK_TITLE, BOOK_COST
FROM BOOK
WHERE BOOK_SUBJECT = 'Database'
ORDER BY BOOK_NUM;
```

67.
```
SELECT CHECK_NUM, BOOK_NUM, CHECK_OUT_DATE
FROM CHECKOUT
WHERE CHECK_OUT_DATE < '2017-04-05'
ORDER BY CHECK_NUM;
```

68.
```
SELECT BOOK_NUM, BOOK_TITLE, BOOK_YEAR
FROM BOOK
WHERE BOOK_YEAR > 2015 AND BOOK_SUBJECT = 'Programming'
ORDER BY BOOK_NUM;
```

69.
```
SELECT BOOK_NUM, BOOK_TITLE, BOOK_SUBJECT, BOOK_COST
FROM BOOK
WHERE (BOOK_SUBJECT = 'Middleware' OR BOOK_SUBJECT = 'Cloud')
AND BOOK_COST > 70
ORDER BY BOOK_NUM;
```

70.
```
SELECT AU_ID, AU_FNAME, AU_LNAME, AU_BIRTHYEAR
FROM AUTHOR
WHERE AU_BIRTHYEAR BETWEEN 1980 AND 1989
ORDER BY AU_ID;
```

71.
```
SELECT BOOK_NUM, BOOK_TITLE, BOOK_SUBJECT
FROM BOOK
WHERE UPPER(BOOK_TITLE) LIKE '%DATABASE%'
ORDER BY BOOK_NUM;
```

72.
```
SELECT PAT_ID, PAT_FNAME, PAT_LNAME
FROM PATRON
WHERE UPPER(PAT_TYPE) = 'STUDENT'
ORDER BY PAT_ID;
```

73.
```
SELECT PAT_ID, PAT_FNAME, PAT_LNAME, PAT_TYPE
FROM PATRON
WHERE UPPER(PAT_LNAME) LIKE 'C%'
ORDER BY PAT_ID;
```

74.
```
SELECT AU_ID, AU_FNAME, AU_LNAME
FROM AUTHOR
WHERE AU_BIRTHYEAR IS NULL
ORDER BY AU_ID;
```

75.
```
SELECT AU_ID, AU_FNAME, AU_LNAME
FROM AUTHOR
WHERE AU_BIRTHYEAR IS NOT NULL
ORDER BY AU_ID;
```

76.
```
SELECT CHECK_NUM, BOOK_NUM, PAT_ID, CHECK_OUT_DATE, CHECK_DUE_DATE
FROM CHECKOUT
WHERE CHECK_IN_DATE IS NULL
ORDER BY BOOK_NUM;
```

77.
```
SELECT AU_ID, AU_FNAME, AU_LNAME, AU_BIRTHYEAR
FROM AUTHOR
ORDER BY AU_BIRTHYEAR DESC, AU_LNAME;
```

78.
```
SELECT COUNT(BOOK_NUM) AS "Number of Books"
FROM BOOK;
```

79.
```
SELECT COUNT(DISTINCT BOOK_SUBJECT) AS "Number of Subjects"
FROM BOOK;
```

80.
```
SELECT COUNT(BOOK_NUM) AS "Available Books"
FROM BOOK
WHERE PAT_ID IS NULL;
```

81.
```
SELECT MAX(BOOK_COST) AS "Most Expensive"
FROM BOOK;
```

82.
```
SELECT MIN(BOOK_COST) AS "Least Expensive"
FROM BOOK;
```

83.
```
SELECT COUNT(DISTINCT PAT_ID) AS "DIFFERENT PATRONS"
FROM CHECKOUT;
```

84.
```
SELECT BOOK_SUBJECT, COUNT(*) AS "Books IN Subject"
FROM BOOK
GROUP BY BOOK_SUBJECT;
```

85.
```
SELECT AU_ID, COUNT(*) AS "Books Written"
FROM WRITES
GROUP BY AU_ID
ORDER BY COUNT(*) DESC, AU_ID;
```

86.
```
SELECT SUM(BOOK_COST) AS "Library Value"
FROM BOOK;
```

87.
```
SELECT PAT_ID AS PATRON, BOOK_NUM AS BOOK, DATEDIFF(CHECK_IN_DATE, CHECK_OUT_DATE) AS "Days Kept"
FROM CHECKOUT
ORDER BY DATEDIFF(CHECK_IN_DATE, CHECK_OUT_DATE) DESC, PAT_ID, BOOK_NUM;
```

88.
```
SELECT PAT_ID, CONCAT(PAT_FNAME, ' ', PAT_LNAME) AS "Patron Name", PAT_TYPE
FROM PATRON
ORDER BY PAT_ID;
```

89.
```
SELECT BOOK_NUM, CONCAT(BOOK_TITLE, ' (', BOOK_YEAR, ')') AS BOOK, BOOK_SUBJECT
FROM BOOK
ORDER BY BOOK_NUM;
```

90.
```
SELECT AU_LNAME, AU_FNAME, BOOK_NUM
FROM AUTHOR JOIN WRITES ON AUTHOR.AU_ID = WRITES.AU_ID
ORDER BY AU_LNAME, AU_FNAME, BOOK_NUM;
```

91.
```
SELECT AU_ID, BOOK.BOOK_NUM, BOOK_TITLE, BOOK_SUBJECT
FROM BOOK JOIN WRITES ON BOOK.BOOK_NUM = WRITES.BOOK_NUM
ORDER BY BOOK.BOOK_NUM, AU_ID;
```

92.
```
SELECT AU_LNAME, AU_FNAME, BOOK_TITLE, BOOK_COST
FROM AUTHOR JOIN WRITES ON AUTHOR.AU_ID = WRITES.AU_ID JOIN BOOK ON WRITES.BOOK_NUM =
BOOK.BOOK_NUM
ORDER BY BOOK.BOOK_NUM, AUTHOR.AU_ID;
```

93.
```
SELECT PATRON.PAT_ID, BOOK_NUM, PAT_FNAME, PAT_LNAME, BOOK_TITLE
FROM PATRON JOIN BOOK ON PATRON.PAT_ID = BOOK.PAT_ID
ORDER BY PAT_LNAME, BOOK_TITLE;
```

94.
```
SELECT PAT_ID, CONCAT(PAT_FNAME, ' ', PAT_LNAME) AS NAME, PAT_TYPE
FROM PATRON
ORDER BY UPPER(PAT_TYPE), UPPER(PAT_LNAME), PAT_FNAME;
```

95.
```
SELECT BOOK_NUM, COUNT(*) AS "Times Checked Out"
FROM CHECKOUT
GROUP BY BOOK_NUM
ORDER BY COUNT(*) DESC, BOOK_NUM DESC;
```

96.
```
SELECT AUTHOR.AU_ID, AU_FNAME, AU_LNAME, BOOK.BOOK_NUM, BOOK_TITLE
FROM AUTHOR JOIN WRITES ON AUTHOR.AU_ID = WRITES.AU_ID JOIN BOOK ON WRITES.BOOK_NUM =
BOOK.BOOK_NUM
WHERE BOOK_SUBJECT = 'Cloud'
ORDER BY BOOK_TITLE, AU_LNAME;
```

97.
```
SELECT BOOK_NUM, BOOK_TITLE, PATRON.PAT_ID, PAT_LNAME, PAT_TYPE
FROM BOOK JOIN PATRON ON BOOK.PAT_ID = PATRON.PAT_ID
ORDER BY BOOK_TITLE;
```

98.
```
SELECT BOOK.BOOK_NUM, BOOK_TITLE, COUNT(CHECK_NUM) AS "Times Checked Out"
FROM BOOK LEFT JOIN CHECKOUT ON BOOK.BOOK_NUM = CHECKOUT.BOOK_NUM
GROUP BY BOOK.BOOK_NUM, BOOK_TITLE
ORDER BY COUNT(CHECK_NUM) DESC, BOOK_TITLE;
```

99.
```
SELECT BOOK.BOOK_NUM, BOOK_TITLE, COUNT(CHECK_NUM) AS "Times Checked Out"
FROM BOOK JOIN CHECKOUT ON BOOK.BOOK_NUM = CHECKOUT.BOOK_NUM
GROUP BY BOOK.BOOK_NUM, BOOK_TITLE
HAVING COUNT(CHECK_NUM) > 5
ORDER BY COUNT(CHECK_NUM) DESC, BOOK_TITLE;
```

100.
```
SELECT AUTHOR.AU_ID, AU_LNAME, BOOK_TITLE, CHECK_OUT_DATE, PAT_LNAME
FROM AUTHOR JOIN WRITES ON AUTHOR.AU_ID = WRITES.AU_ID JOIN BOOK ON WRITES.BOOK_NUM =
BOOK.BOOK_NUM
JOIN CHECKOUT ON BOOK.BOOK_NUM = CHECKOUT.BOOK_NUM JOIN PATRON ON PATRON.PAT_ID =
CHECKOUT.PAT_ID
WHERE PAT_LNAME = 'Miles' AND AU_LNAME = 'Bruer'
ORDER BY CHECK_OUT_DATE;
```

101.
```
SELECT PATRON.PAT_ID, PAT_FNAME, PAT_LNAME
FROM PATRON LEFT JOIN CHECKOUT ON PATRON.PAT_ID = CHECKOUT.PAT_ID
WHERE CHECK_NUM IS NULL
ORDER BY PAT_LNAME, PAT_FNAME;
```

102.
```
SELECT PATRON.PAT_ID, PAT_LNAME, COUNT(CHECK_NUM) AS "NUM CHECKOUTS", COUNT(DISTINCT BOOK_NUM) AS
"NUM DIFFERENT BOOKS"
FROM CHECKOUT JOIN PATRON ON CHECKOUT.PAT_ID = PATRON.PAT_ID
GROUP BY PATRON.PAT_ID, PAT_LNAME
HAVING COUNT(CHECK_NUM) > 2
ORDER BY COUNT(DISTINCT BOOK_NUM) DESC, COUNT(CHECK_NUM) DESC, PATRON.PAT_ID;
```

103.
```
SELECT ROUND(AVG(DATEDIFF(CHECK_IN_DATE, CHECK_OUT_DATE)), 2) AS "Average Days Kept"
FROM CHECKOUT;
```

104.
```
SELECT PAT_ID, ROUND(AVG(DATEDIFF(CHECK_IN_DATE, CHECK_OUT_DATE)), 2) AS "Average Days Kept"
FROM CHECKOUT
GROUP BY PAT_ID
HAVING COUNT(CHECK_NUM) > 2
ORDER BY ROUND(AVG(DATEDIFF(CHECK_IN_DATE, CHECK_OUT_DATE)), 2) DESC;
```

105.
```
SELECT BOOK_NUM, BOOK_TITLE, BOOK_COST
FROM BOOK
WHERE BOOK_COST = (SELECT MIN(BOOK_COST) FROM BOOK)
ORDER BY BOOK_NUM;
```

106.
```
SELECT AU_ID, AU_FNAME, AU_LNAME
FROM AUTHOR
WHERE AU_ID NOT IN (SELECT AU_ID FROM BOOK JOIN WRITES ON BOOK.BOOK_NUM =  WRITES.BOOK_NUM WHERE
BOOK_SUBJECT = 'Programming')
ORDER BY AU_LNAME;
```

107.
```
SELECT BOOK_NUM, BOOK_TITLE, BOOK.BOOK_SUBJECT, ROUND(AVGCOST, 2) AS "Average Subject Cost", BOOK_COST -
ROUND(AVGCOST, 2) AS DIFFERENCE
FROM BOOK JOIN (SELECT BOOK_SUBJECT, AVG(BOOK_COST) AS AVGCOST
FROM BOOK BOOK2
GROUP BY BOOK_SUBJECT) AS SUBAVGS ON BOOK.BOOK_SUBJECT = SUBAVGS.BOOK_SUBJECT
ORDER BY BOOK_TITLE;
```

108.
```
SELECT BOOK.BOOK_NUM, BOOK_TITLE, BOOK_SUBJECT, AU_LNAME, NUMBOOKS AS "Num Books by Author"
FROM BOOK JOIN WRITES ON BOOK.BOOK_NUM = WRITES.BOOK_NUM JOIN
(SELECT AUTHOR.AU_ID, AU_LNAME, COUNT(*) AS NUMBOOKS
FROM AUTHOR JOIN WRITES ON AUTHOR.AU_ID = WRITES.AU_ID
GROUP BY AUTHOR.AU_ID, AU_LNAME) AS AUTHBOOKS ON WRITES.AU_ID = AUTHBOOKS.AU_ID
WHERE BOOK_SUBJECT = 'Cloud'
ORDER BY BOOK_TITLE, AU_LNAME;
```

109.
```
SELECT MIN(AVGCOST) AS "Lowest AVG Cost", MAX(AVGCOST) AS "Highest AVG Cost"
FROM (SELECT BOOK_SUBJECT, ROUND(AVG(BOOK_COST), 2) AS AVGCOST
FROM BOOK
GROUP BY BOOK_SUBJECT) AS SUBAVGS;
```

Part 2

1.
```
CREATE TABLE EMP_1 (

EMP_NUM         CHAR(3)          NOT NULL   UNIQUE,
EMP_LNAME       VARCHAR(15)      NOT NULL,
EMP_FNAME       VARCHAR(15)      NOT NULL,
EMP_INITIAL     CHAR(1),
EMP_HIREDATE    DATE,
JOB_CODE        CHAR(3),
CONSTRAINT EMP1_EMP_NUM_PK PRIMARY KEY (EMP_NUM),
CONSTRAINT EMP1_JOB_CODE_FK FOREIGN KEY (JOB_CODE) REFERENCES JOB (JOB_CODE)
);
```

2.
```
INSERT INTO EMP_1 VALUES ('101', 'News', 'John', 'G', '08-Nov-00', '502');

INSERT INTO EMP_1 VALUES ('102', 'Senior', 'David', 'H', '12-Jul-89', '501');
```

4.
```
COMMIT ;
```

5.
```
UPDATE EMP_1
SET JOB_CODE = '501'
WHERE   EMP_NUM = '107';
```

6.
```
 DELETE  FROM EMP_1
WHERE   EMP_LNAME = 'Smithfield'
AND EMP_FNAME = 'William' AND   EMP_HIREDATE = '22-June-04' AND JOB_CODE = '500';
```

7.
```
SELECT * INTO EMP_2
FROM EMP_1;
```

8.
```
ALTER TABLE EMP_2
ADD EMP_PCT     NUMERIC (4,2),
PROJ_NUM    CHAR(3);
```

9.
```
UPDATE EMP_2
SET EMP_PCT = 3.85
WHERE EMP_NUM = '103';
```