CS525-01/02/03 - Spring 2018
Problem Set 3
Due in Blackboard on Tuesday, April 10th (11:59pm Chicago Time)
Handwritten/Scanned submission is accepted.

---

**Problem 1. 6 Points** (2 Points each): Suppose blocks of 1024 bytes are used to store variable-length records. The fixed part of the block header is 24 bytes, and includes a count of the number of records. In addition, the header contains a 4-byte pointer (offset) to each record in the block. The first record is placed at the end of the block, the second record starts where the first one ends, and so on. The size of each record is not stored explicitly, since the size can be computed from the pointers.

1.1. How many records of size 50 bytes could we fully fit (no spanning) in such a block? (The block is designed to hold variable size records, but these records by coincidence happen to be all the same size.)

$$Number\ of\ records = \left\lfloor \frac{1,024 - 24}{50 + 4} \right\rfloor = 18$$

1.2. Suppose we want to fully store 20 records in the block, again all of the same size. What is the maximum size of such records?

$$Maximum\ record\ size = \left\lfloor \frac{1,024 - 24 - 4 \times 20}{20} \right\rfloor = 46$$

1.3. Now assume the block layout is changed to accommodate only fixed size records. In this case, the block header is still 24 bytes and contains the common length of the records in the block. How many records of size 50 bytes could we fully fit in such a block?

$$Number\ of\ records = \left\lfloor \frac{1,024 - 24}{50} \right\rfloor = 20$$

1.4. For this part, assume we allow record spanning. Records are fixed size, but since we can have record fragments we need to add a 5-byte header to every fragment to store its length (and other information). (The block header is still 24 bytes.)
Suppose we have 2 blocks of 1024 bytes each, we would like to store 3 records of 600 bytes each. We first allocate as much as we can of the records into the first block, and then use space from the second block. After the 3 records are stored, how many bytes are still available in the second block?
Free bytes available: 180
Record 2 is split into fragments 2-a and 2-b. Block 1 has room for 390 bytes of record 2-a (1024-24-5-600-5). Block 2 needs to store 210 bytes of record 2-b. Block 2 also needs to store block header, record 3, and two record headers (for record 2-b and 3). Therefore, 180 bytes (1024-24-5-210-5-600) are still available in block 2.

**Problem 2. 8 Points** (2 Points each): You have been hired to work on a web site that maintains customer reviews of products. The main data is stored in the following tables:

- `Product`(`pid`:**string**, `pname`:**string**, `description`:**string**)
- `Reviewer`(`rid`:**string**, `rname`:**string**, `city`:**string**)
- `Review`(`rid`:**string**, `pid`:**string**,`rating`:**integer**, `comment`:**string**)

The tables contain the following information:

- `Product`: unique product id (`pid`), product name (`pname`), and product description. All strings.
- `Reviewer`: unique reviewer id (`rid`), and reviewer name (`rname`) and `city`, also all strings.

– `Review`: One entry for each individual review giving the reviewer and product ids, an integer `rating` in the range `0-5`, and the reviewer comment, which is a string.

Write the following queries in `SQL`. No duplicates should be printed in any of the answers.

2.1 Return the ratings and comments from all reviews written by reviewers in Chicago for products named 'ABC'. The results should be sorted in descending order by rating.

```
SELECT r.rating, r.comment
FROM Product p, Reviewer rr, Review r
WHERE p.pname = ABC AND rr.city = Chicago AND p.pid = r.pid AND rr.rid = r.rid
ORDER BY r.rating DESC
```

2.2 Return the number of reviewers in each distinct city. The results should list the city name and the number of reviewers in that city, and should be sorted alphabetically by city name.

```
SELECT rr.city, COUNT(rr.rid)
FROM Reviewer rr
GROUP BY rr.city
ORDER BY rr.city
```

2.3 Return the names of all grumpy reviewers. A grumpy reviewer is one whose average review rating is less than or equal to 2. If someone is in the `Reviewer` table but has no reviews in the `Review` table, they should not be included in the result. The reviewer names in the result can be in any order.

```
SELECT rr.rname
FROM Reviewer rr, Review r
WHERE rr.rid = r.rid
GROUP BY r.rid, rr.rid, rr.rname
HAVING AVG(r.rating) <= 2
```

2.4 Return the names and descriptions of all cool products. A "cool product" is one that has no reviews in the database with any rating less than 4. A product must have at least one review in the database to be considered as a possible "cool product". The results can be in any order.

```
SELECT p.pname, p.description
FROM Product p, Review r
WHERE p.pid = r.pid
GROUP BY p.pid, p.pname, p.description
HAVING MIN(r.rating) >= 4
```

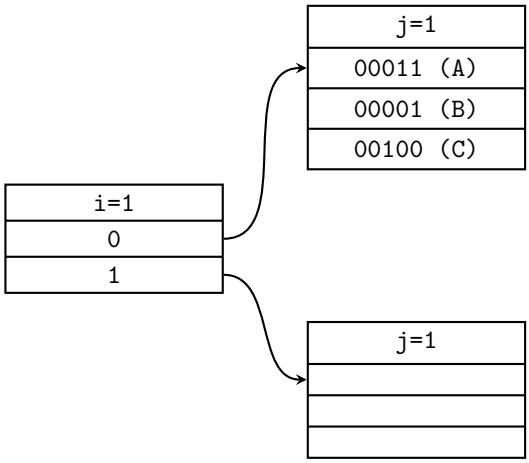
**Problem 3. 15 Points** : Consider an `extensible hash` structure where buckets can hold up to three entries. Initially the structure is empty. The keys and their hash values are the following.

| $x$ | $h(x)$ |
|---|---|
| A | 00011 |
| B | 00001 |
| C | 00100 |
| D | 00000 |
| E | 00001 |
| F | 00010 |
| G | 00011 |
| H | 00101 |
| I | 00110 |
| J | 00101 |
| K | 00001 |
| L | 00110 |

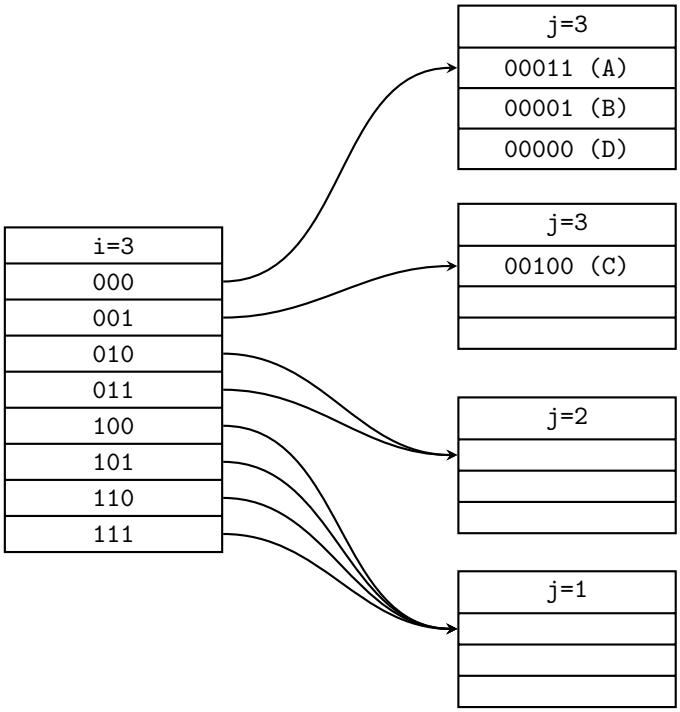
Suppose we insert the values in the order given above. Show the structure after all the values have been inserted.

Answer

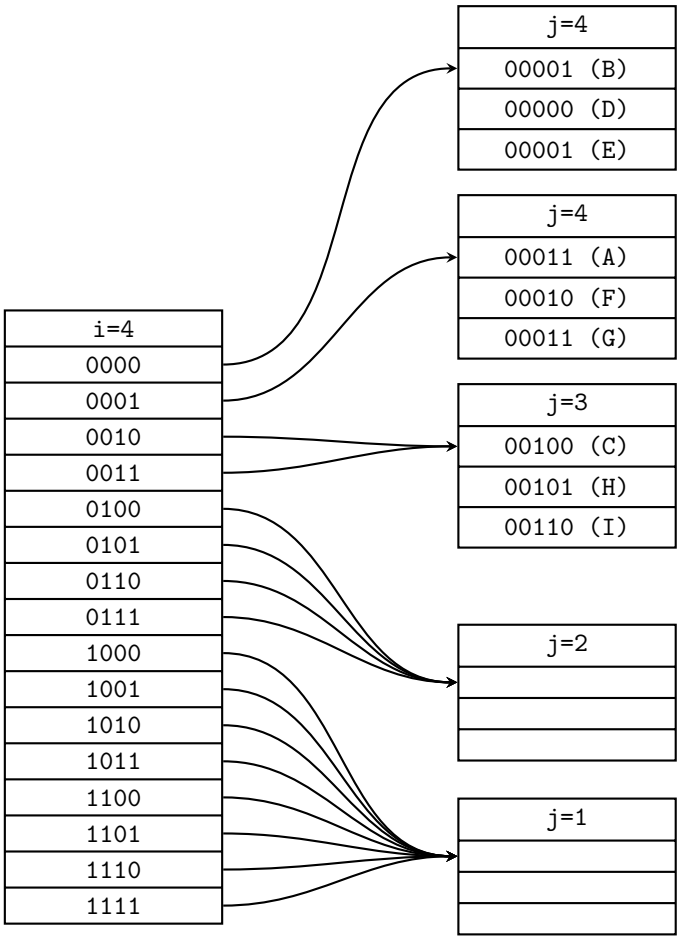
insert(A), (B), (C)



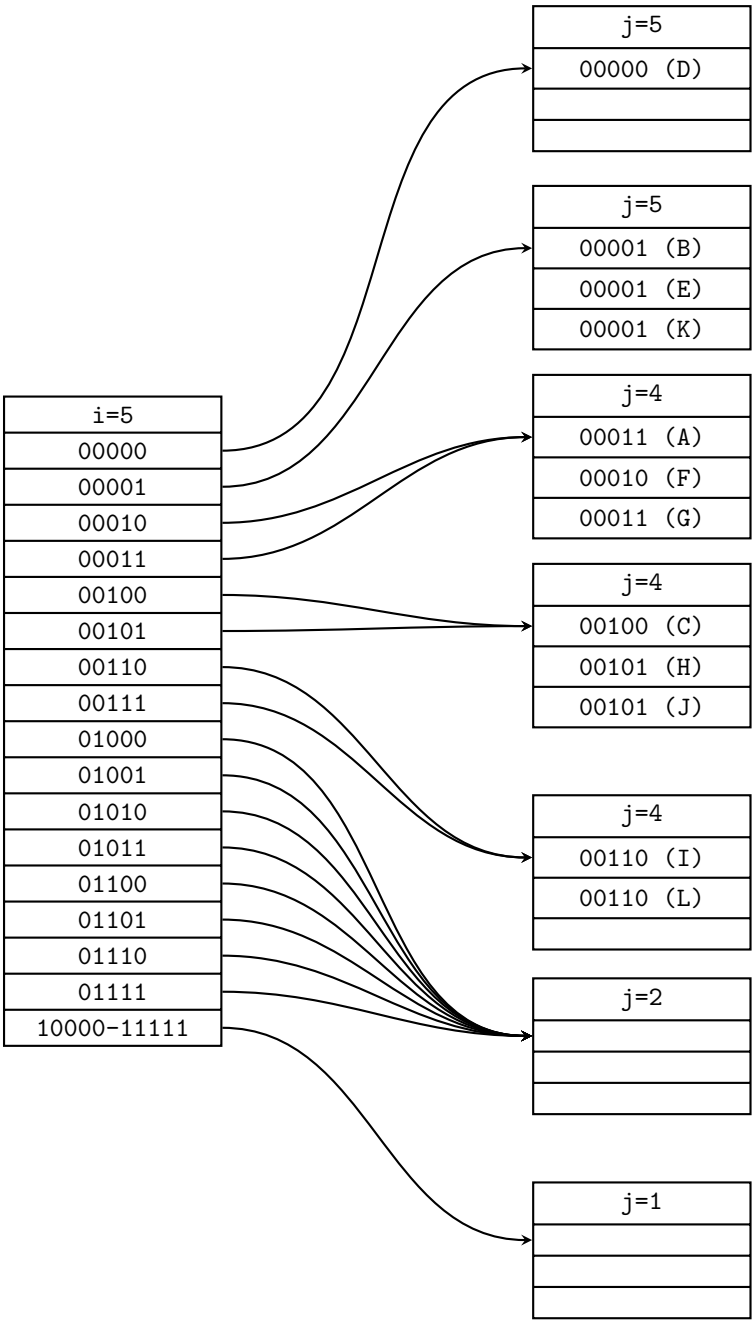

insert(D)



insert(E), then (F), (G), (H), (I)



insert(J), then (K), (L)




**Problem 4. 18 Points** (a) 2 Points, b) 4 Points each): Consider an **extensible hash** structure with the following characteristics:

- Buckets can hold up to two records.
- No overflow blocks are allowed.
- The hash function we use generates `b = 4` bits total.
- Initially the extensible hash table is empty.

Say we insert `X` records, where the search key of each record generates a distinct 4-bit hash value (no

collisions). No records are deleted during this process. We are told that after the `X` insertions, `4` buckets have been allocated. (Note that the previous sentence does not refer to the size of the directory.)

4.1. What is the minimum possible value of `X`?
Minimum value of X is 3

4.2. In the same scenario, what is the maximum possible value of `X`?
Maximum value of X is 8

**Problem 5. 10 Points** `(2.5 Points each)`: Suppose we have blocks of size 1024 bytes that we use to store fixed-length records. Each block has a 32 byte header used to store information including the number of records in the block.

(a) Suppose we have records consisting of a 12 byte header, and 3 fields of size 5 bytes, 6 bytes and 7 bytes respectively. Within each record, fields can start at any byte. How many records can we fit in a block?
$1024 - 32 = 992$ bytes available for records in each block

$(12 + 5 + 6 + 7) = 30$ bytes per record

$\frac{992}{30} = 33$ records per block

(b) Now consider blocks of size `1024` bytes that we use to store variable-length records. Each block has a fixed `32` byte header used to store information including the number of records in the block. In addition to this fixed header, the header contains variable number of `2` byte pointers to each record in the block. Records can start at any byte offset and are packed as densely as possible. Which of these following combinations of records can be stored in a single block? Check all that apply. Show your work.

☑ `32` records of `23` bytes each
32 records of 23 bytes each
32 records require 23*32 bytes of space. 1024-32-2*32-23*32=192 bytes free

☑ `30` records of 20 bytes each and `10` records of `10` bytes each
40 records total
1024-32-2*40-30*20-10*10 = 212 bytes free

☑ `5` records of `11` bytes, `10` records of `13` bytes and `20` records of `17` bytes
35 records total
1024-32-2*35-5*11-10*13-20*17 = 397 bytes free

☒ `2` records of size `496`
2 records total
1024-32-2*2-496*2 = -2 (out of space!)

**Problem 6. 4 Points** : Give examples to show that:

6.1. Projection cannot be pushed below set union.
Let R(A,B) = {(1,2)} and S(A,B) = {(1,3)}. Then $\pi_A(R \cup_s S) = \pi_A(\{(1,2),(1,3)\} = \{(1),(1)\}$. But $\pi_A(R) \cup_s \pi_B(S) = \{(1)\}$, since the set union removes duplicates from the union.

6.2. Projection cannot be pushed below set or bag difference.
R(a,b,c)={(1,2,3), (1,3,2)} and S(a,b,c)={(1,2,3), (1,4,5)} provide an example which proves the assertion

6.3. Duplicate elimination cannot be pushed below projection.
R(a,b,c)={(1,2,3), (1,3,2)} and S(a,b,c)={(1,2,3), (1,4,5)} provide an example which proves the assertion

6.4. Duplicate elimination cannot be pushed below bag union or difference.
R(a)={(1),(1)} and S(a)={(1),(1)} provide an example which proves the assertion.

**Problem 7. 10 Points** `(2 Points each)`: Write the relational algebra expressions for Part 1.2 in Quiz I

1.2.1.

$$\pi_{sname}\big(\pi_{sid}((\pi_{pid}\sigma_{color='red'}\mathbf{Parts}) \bowtie \mathbf{Catalog}) \bowtie \mathbf{Suppliers}\big)$$

1.2.2.

$$\pi_{sid}\big(\pi_{pid}(\sigma_{color='red' \vee color='green'}\mathbf{Parts}) \bowtie \mathbf{Catalog}\big)$$

1.2.3.

$$\rho\big(R1 \leftarrow \pi_{sid}((\pi_{pid}\sigma_{color='red'}\mathbf{Parts}) \bowtie \mathbf{Catalog})\big)$$
$$\rho(R2 \leftarrow \pi_{sid}\sigma_{address='10thWestStreet'}\mathbf{Suppliers})$$
$$R1 \cup R2$$

1.2.4.

$$\rho\big(R1 \leftarrow \pi_{sid}((\pi_{pid}\sigma_{color='red'}\mathbf{Parts}) \bowtie \mathbf{Catalog})\big)$$
$$\rho\big(R2 \leftarrow \pi_{sid}((\pi_{pid}\sigma_{color='green'}\mathbf{Parts}) \bowtie \mathbf{Catalog})\big)$$
$$R1 \cap R2$$

1.2.5.

$$\rho(R1 \leftarrow \mathbf{Catalog})$$
$$\rho(R2 \leftarrow \mathbf{Catalog})$$
$$\pi_{R1.sid,R2.sid}\big(\sigma_{R1.pid=R2.pid \wedge R1.sid \neq R2.sid \wedge R1.cost > R2.cost}(R1 \times R2)\big)$$

**Problem 8. 15 Points** `(3 Points each except 8.4) 6 Points)`: Consider a table book with attributes ISBN, title, author, edition (primary key is ISBN), a table library with loc, budget, public (primary key is loc), and a table catalog with attributes library and book. catalog.library is a foreign key to relation library. Attribute book of relation catalog is a foreign key to relation book. Given are the following statistics:

$$T(book) = 100,000 \qquad T(library) = 100 \qquad T(catalog) = 200,000$$
$$V(book, ISBN) = 100,000 \qquad V(library, loc) = 100 \qquad V(catalog, library) = 100$$
$$V(book, title) = 50,000 \qquad V(library, budget) = 40 \qquad V(catalog, book) = 90,000$$
$$V(book, author) = 30,000 \qquad V(library, public) = 2$$
$$V(book, edition) = 15$$

8.1. Estimate the number of result tuples for the query $q = \sigma_{public=true}(library)$ using the first assumption presented in class (values used in queries are uniformly distributed within the active domain).

$$T(q) = \frac{T(library)}{V(library, public)} = \frac{100}{2} = 50$$

8.2. Estimate the number of result tuples for the query $q = \sigma_{title=Faust \wedge author=Goethe}(book)$ using the first assumption presented in class.

$$T(q) = \frac{T(book)}{V(book, title) \cdot V(book, author)} = \frac{100,000}{50,000 \cdot 30,000} = \frac{1}{15,000}$$

8.3. Estimate the number of result tuples for the query $q = \sigma_{edition \geq 2 \wedge edition \leq 4 \wedge title=Databases}(book)$ using the first assumption presented in class. Assume that the minimal and maximal values in the `edition` attribute are 1 and 15

$$T(q) = \frac{4 - 2 + 1}{max(book, edition) - min(book, edition) + 1} \cdot \frac{1}{50,000} \cdot T(book) = \frac{3 \cdot 100,000}{15 \cdot 50,000} = \frac{2}{5} = 0.4$$

8.4. Estimate the number of result tuples for the query $q = \sigma_{title=DatabaseIntroduction \vee title=DatabaseSystems}(book) \bowtie_{title=book} catalog \bowtie_{library=loc} \sigma_{budget \leq 40}(library)$ using the first assumption presented in class. Assume that the minimal and maximal values in the `budget` attribute are 10 and 70

**We will compute the size estimates without using information about foreign keys**
Let $q_1 = \sigma_{title=DatabaseIntroduction \vee title=DatabaseSystems}(book)$ and $q_2 = \sigma_{budget \leq 40}(library)$.
To estimate the selection result size $q_1$:

$$T(q_1) = (1 - [(1 - \frac{1}{V(book, title)}) \cdot (1 - \frac{1}{V(book, title)})]) \cdot T(book)$$
$$= (1 - (1 - \frac{1}{50,000}) \cdot (1 - \frac{1}{50,000})) \cdot 10,000 \approx 4$$

To estimate the selection result size $q_2$:

$$T(q_2) = \frac{max(library, budget) - 40 + 1}{max(library, budget) - min(library, budget) + 1} \cdot T(library)$$
$$= \frac{70 - 40 + 1}{70 - 10 + 1} \cdot 100 = \frac{31}{61} \cdot 100 \approx 51$$

$$Now\ T(\sigma_{title=DatabaseIntroduction \vee title=DatabaseSystems}(book) \bowtie_{title=book} catalog)$$
$$= T(q_1 \bowtie_{title=book} catalog) = \frac{T(q_1) \times T(catalog)}{max\big(V(q_1, title), V(catalog, name)\big)}$$
$$= \frac{4 \times 200,000}{max\big(4, 90,000\big)} = \frac{80}{9} \approx 8.89$$

Now for the full query we get

$$T(q) = \frac{T(q_1 \bowtie_{title=book} catalog) \times T(q_2)}{max(V(q_1 \bowtie_{title=book} catalog, library), V(q_2, loc))}$$

$$= \frac{8.89 \times 51}{max(100, 51)} =\approx 4.53$$