1.

(a): =16 records =625 blocks

(b):

(c): size=4+8+4+6+4=26 bytes

blocks

(d): 1103 blocks

(e): 10000\*4+8\*18\*10000=1480000 bytes

2.

(a): 625\*1=625ms

(b): (Log2(30)+1)ms

(c): 1147\*1=1147ms

(d): Log2(1103)+1ms

(e): Log2(362)+1826.5ms

(f): Log2(1103)+1

3.

• Find records with search-key value 11.

(a) 4+1=5

(b) 2+1=3

• Find records with search-key values between 5 and 19 (inclusive).

(a) 4+9=13

(b) 2+2+5=9

• Find records with search-key values less than 17.

(a) 4+9=13

(b) 2+2+5=9

4. Integer is 4 Bytes, so R1=6 bytes and R2=6 bytes.

• SELECT \* FROM R1 WHERE b=99577

blocks=100000/150=667

• SELECT \* FROM R2 WHERE c=512213

4+1=5 blocks

• SELECT \* FROM R2 WHERE d=98988 assuming there are 15 records with d = 98988

3+15=18 blocks

• SELECT \* FROM R1 JOIN R2 ON R1.c = R2.c using indexed nested-loop join

Block reads = O(B1 + B1×R1×(H2 + M))=667+100000\*(4)=400667

• SELECT \* FROM R1 JOIN R2 ON R1.c = R2.c using block nested-loop join

Block reads = 667+667\*225=150742

• SELECT \* FROM R1 JOIN R2 ON R1.c = R2.c using sort-merge join

O(2B1log2B1+ B1+ B2)= 12515+667+225=13407

5.

(a) SELECT \* FROM R WHERE a=87555 AND b=60155.76

**1. 2+1=3 blocks**

**2. 2+5=7 blocks**

**3. 25000/25=1000 blocks**

(b) SELECT \* FROM R WHERE c=’abc’

**1. 1000 blocks**

**2. 1000 blocks**

**3. 5+1=6 blocks**

(c) SELECT \* FROM R WHERE a BETWEEN 10000 AND 11000 AND c=’abc’

**1. 2+40=42 blocks**

**2. 1000 blocks**

**3. 5+1=6 blocks**

(d) SELECT \* FROM R WHERE a BETWEEN 10000 AND 11000 AND b=60155.76

**1. 42 blocks**

**2. 2+5=7 blocks**

**3. 25000/25=1000 blocks**

(e) SELECT \* FROM R WHERE a BETWEEN 10000 AND 11000 OR c=’abc’

**By using both a and c, so find the records between 10000 and 11000 cost us 42 blocks, find the records c=’abc’ cost 6 blocks, which returns a total block access time as 48 blocks to find the result.**

6.

Part1:

• πname (σdept name=’Comp. Sci. (Student))

To select students’ names, which are in computer science department.

**select name from student where dept\_name = "CS";**

• πname (Student ◃▹ Takes ◃▹ σdept name=’Physics’ (Course))

To select students’ names using joins, which take courses from physics department.

**select distinct student.name from student,takes,course where student.ID = takes.ID and course.course\_id = takes.course\_id and course.dept\_name = "Physics";**

• πcourse id (σsemester=’Spring’∧year=2016 (Section))∩πcourse id (σsemester=’Fall’∧year=2015 (Section)) • dept nameGaverage(salary) (πdept name,salary (Instructor))

To select these courses’ course\_id from Section table, which are taught both in Spring 2016 and Fall 2015.

**select a.course\_id from section a join section b on a.course\_id = b.course\_id where a.semester="fall" and a.year="2015" and b.semester="spring" and b.year="2016";**

• dept nameGaverage(salary) (πdept name,salary (Instructor))

To show all departments and their instructors’ average salary, the result is grouped by dept\_name.

**select dept\_name,avg(salary) from instructor group by dept\_name;**

• nameGcount(s id) (πname, s id (σdept name=’Comp. Sci.’ (Instructor) ◃▹ID = i id Advisor))

First, this one chooses advisors(instructor) in computer science department, and then chooses these advisors’ students(s\_id) and these instructors’ names. Finally, it use aggregation function count() to calculate the total amount of students for each advisor.

**select instructor.name,count(advisor.s\_id) as std\_count from advisor,instructor where instructor.ID=advisor.i\_id and instructor.dept\_name="CS" group by instructor.name**;

Part2:

• What are the titles of courses taught by Professor Wu?

select course.title from course,teaches,instructor where teaches.ID=instructor.ID and instructor.name="Wu" and course.course\_id=teaches.course\_id; **πtitle(course**⋈course.course\_id=teaches.course\_idteaches⋈teaches.ID=instructor.IDσname=’Wu’(instructor))

• What are the IDs and names of students taking CISC637 in the Fall of 2016?

select student.ID,student.name from student,takes where student.ID=takes.ID and takes.course\_id="CISC637" and takes.semester="fall" and takes.year=2016; **πID,name(**σcourse\_id=’CISC637’∧semester=’fall’∧year=2016**(**takes⋈student))

• What are the IDs and names of instructors who have not taught any course since Fall 2014?

select distinct instructor.ID,instructor.name from instructor,teaches where instructor.ID=teaches.ID and(teaches.year<2014) or (semester='spring' and year=2014);

**πID,name((**Instructor⋈teaches)-σteaches.year>2014(Instructor⋈teaches))

• Find the number of open seats in CISC637 in Fall of 2016.

select (classroom.capacity-count(distinct takes.ID)) as cisc637\_seats from section,classroom,takes where takes.course\_id='CISC637' and takes.sec\_id='015' and takes.semester='fall' and takes.year=2016 and section.course\_id='CISC637' and section.sec\_id='015' and section.building = classroom.building and section.room\_no =classroom.room\_no and section.semester = 'fall' and section.year=2016 group by classroom.capacity;

**capacityG(capacity-count(ID))(πcapacity,ID(**σcourse\_id=’CISC637’∧sec\_id=’015’∧semester=’fall’∧year=’2016’**(takes)**⋈section⋈classroom))