Indian Institute of Information Technology Vadodara

CS262: Database Management System

Lab 4

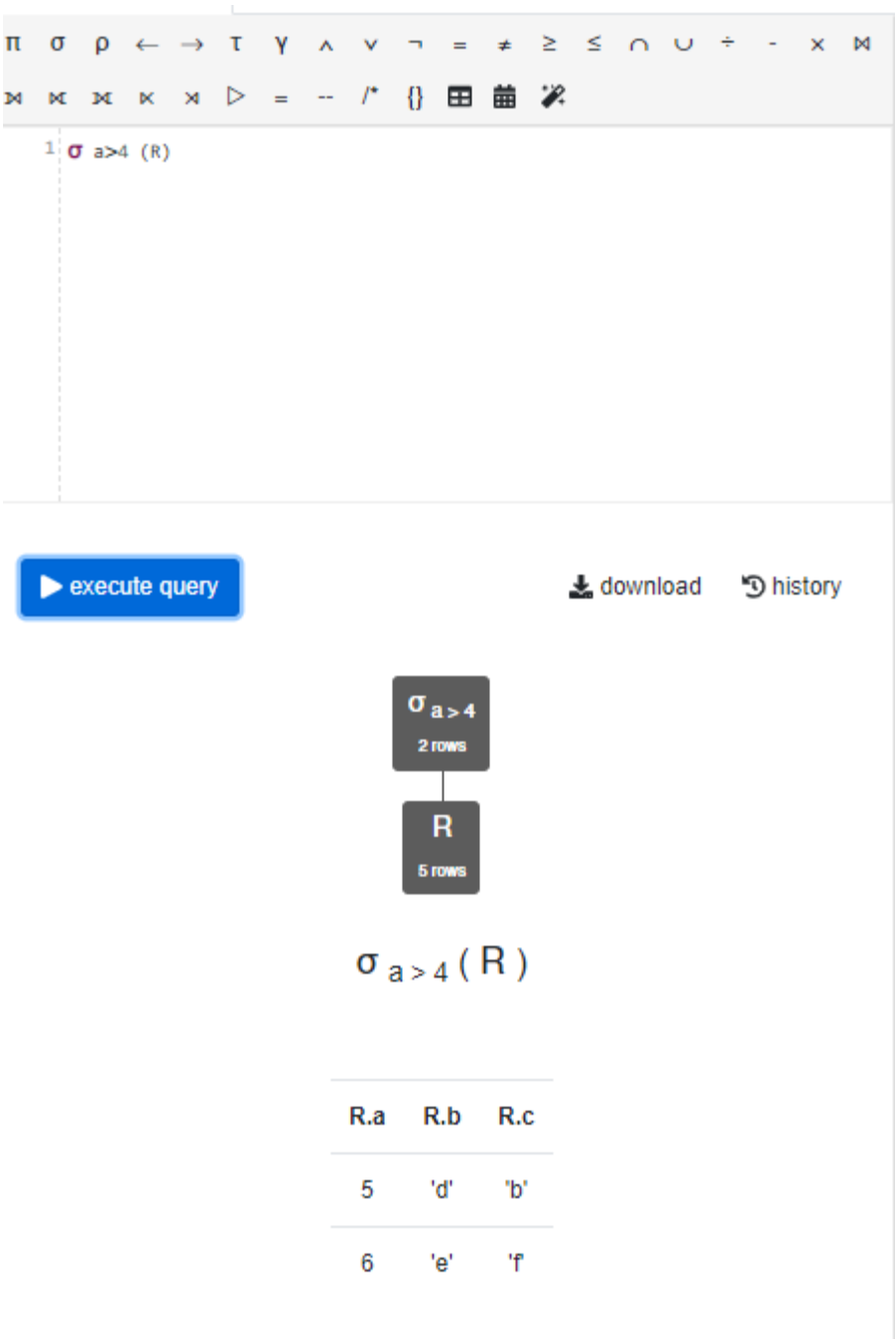
Roll No. 201951105 **Name: Nishant Andoriya**

In this lab we have to use the online tool

(https://dbisuibk.github.io/relax/landing**)**

Using (UIBK – R, S, T) database

Q.1) Find all the tuples in the R with value of attribute ‘a’ greater than 4.

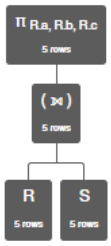


Q.2) Find all the tuples of R that are having the same value in attribute ‘b’ of S.

```
1 π R.a , R.b,R.c (R ⋈ S)
```

▶ execute query

download history



$$\pi_{R.a,\ R.b,\ R.c}\ (R \bowtie S)$$

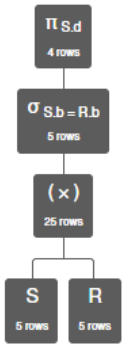
R.a	R.b	R.c
1	'a'	'd'
3	'c'	'c'
4	'd'	'f'
5	'd'	'b'
6	'e'	'f'

Q.3) Find all the values of ‘d’ from S that is having the matching value in attribute ‘b’ of S with the attribute ‘b’ of R

```
1 π S.d (σ S.b= R.b (S× R))
```

execute query

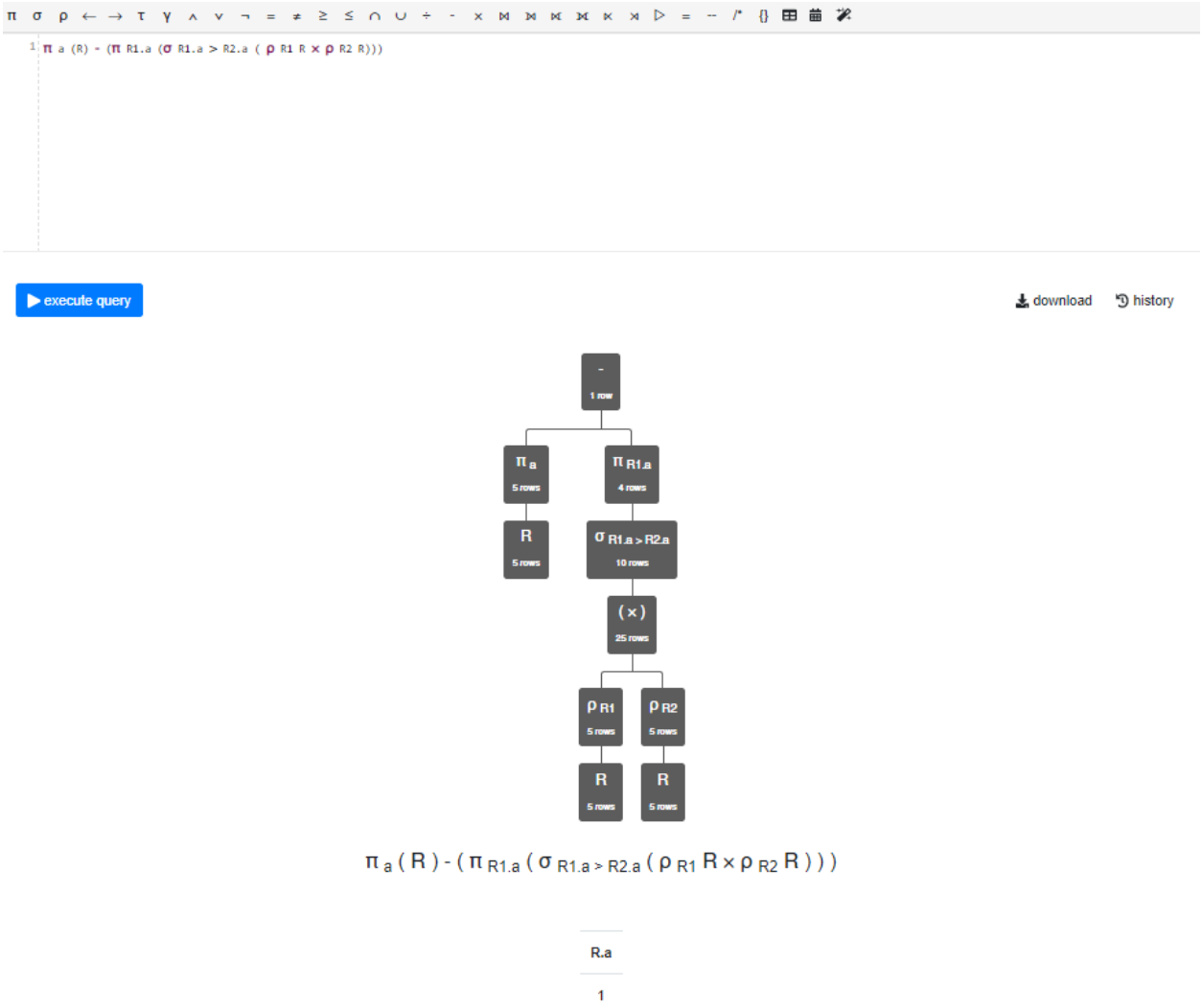
download history



$$\pi_{S.d}(\sigma_{S.b=R.b}(S \times R))$$

S.d
100
400
200
150

Q.4) Find the minimum value of attribute ‘a’ in R.



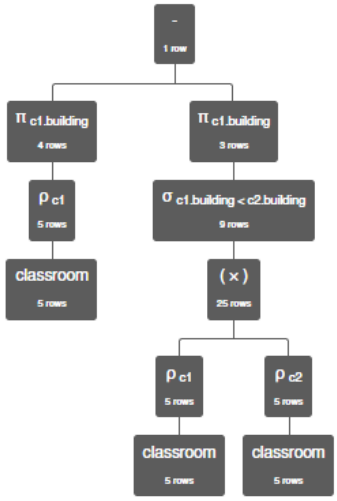
Using Silberschatz-University database

Q.1) Find the building name with the maximum room capacity.

```
π c1.building (ρ c1 classroom) - π c1.building (σ c1.building< c2.building (ρ c1 classroom × ρ c2 classroom ))
```

▶ execute query

download history



$$\pi_{c1.building}(\rho_{c1}\ classroom) - \pi_{c1.building}(\sigma_{c1.building < c2.building}(\rho_{c1}\ classroom \times \rho_{c2}\ classroom))$$

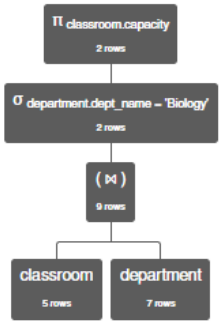
c1.building
'Watson'

Q.2) Find the room capacity of ‘Biology’ department.

```
1 π classroom.capacity (σ department.dept_name = 'Biology' (classroom ⋈ department))
```

execute query

download history



π classroom.capacity (σ department.dept_name = 'Biology' (classroom ⋈ department))

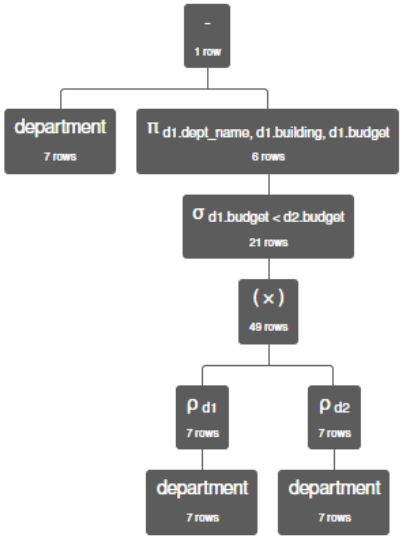
classroom.capacity
30
50

Q.3) Find the department with maximum budget.

```
(department) - (π d1.dept_name, d1.building, d1.budget (σ d1.budget<d2.budget (ρ d1 department × ρ d2 department)))
```

execute query

download history



(department) - (π d1.dept_name, d1.building, d1.budget (σ d1.budget < d2.budget (ρ d1 department × ρ d2 department)))

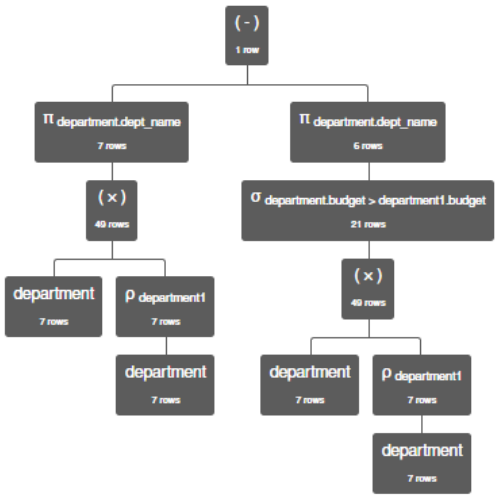
department.dept_name	department.building	department.budget
'Finance'	'Painter'	120000

Q.4) Find the department with minimum budget.

```
1 (π department.dept_name (department  × ρ department1 department) - π department.dept_name (σ department.budget>department1.budget (department × ρ department1 (department))))
```

execute query

download history



(π department.dept_name (department × ρ department1 department) - π department.dept_name (σ department.budget > department1.budget (department × ρ department1 (department))))

department.dept_name
'History'

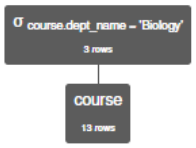
Q.5) Find all the courses related with the ‘Biology’ department.

π σ ρ ← → τ γ ∧ ∨ ¬ = ≠ ≥ ≤ ∩ ∪ ÷ - × ⋈ ⋈ ⋉ ⋊ ⋉ ⋊ ▷ = -- /* {}

```
1 σ course.dept_name = 'Biology' (course)
```

▶ execute query

download history



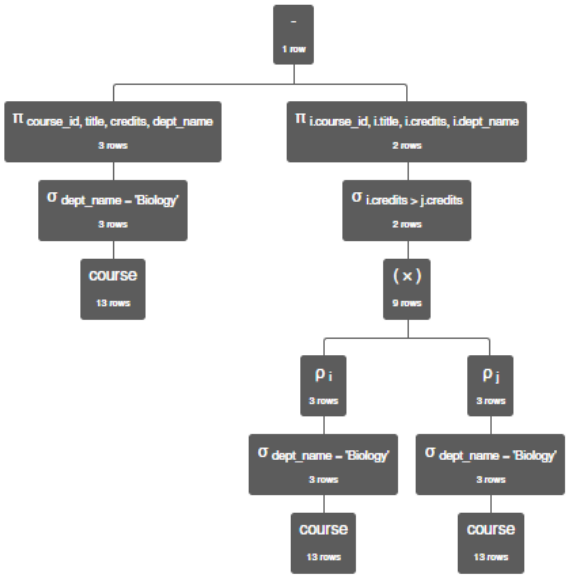
σ course.dept_name = 'Biology' (course)

course.course_id	course.title	course.dept_name	course.credits
'BIO-101'	'Intro. to Biology'	'Biology'	4
'BIO-301'	'Genetics'	'Biology'	4
'BIO-399'	'Computational Biology'	'Biology'	3

Q.6) Find all the courses in the ‘Biology’ department with minimum credits. (For example minimum credit asociated with any course in the biology department is 2 then the answer should include all the courses having credit value equal to 2)

```
π σ ρ ← → τ γ ∧ ∨ ¬ = ≠ ≥ ≤ ∩ ∪ ÷ - × ⋈ ⋈ ⋉ ⋊ ⋉ ⋊ ▷ = -- /* {}
1 π course_id,title,credits,dept_name ( σ dept_name ='Biology' (course)) - π i.course_id,i.title,i.credits,i.dept_name ( σ i.credits> j.credits ((ρ i (σ dept_name ='Biology' (course)))
  × (ρ j (σdept_name ='Biology' (course))) ))
```

▶ execute query download history



$$\pi_{course\_id,\ title,\ credits,\ dept\_name}\left(\sigma_{dept\_name\ =\ 'Biology'}\left(course\right)\right) - \pi_{i.course\_id,\ i.title,\ i.credits,\ i.dept\_name}\left(\sigma_{i.credits\ >\ j.credits}\left(\left(\rho_{i}\left(\sigma_{dept\_name\ =\ 'Biology'}\left(course\right)\right)\right) \times \left(\rho_{j}\left(\sigma_{dept\_name\ =\ 'Biology'}\left(course\right)\right)\right)\right)\right)$$

course.course_id	course.title	course.credits	course.dept_name
'BIO-399'	'Computational Biology'	3	'Biology'

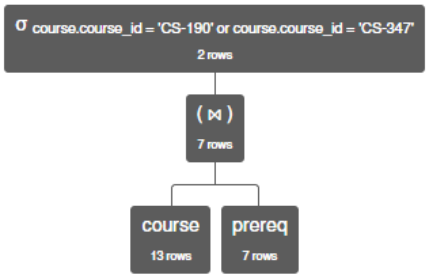
Q.7) Find all the prerequisit courses for the course with ID CS-190

and CS-347.

```
σ course.course_id='CS-190' ∨ course.course_id='CS-347' (course ⋈ prereq)
```

execute query

download history



σ course.course_id = 'CS-190' or course.course_id = 'CS-347' (course ⋈ prereq)

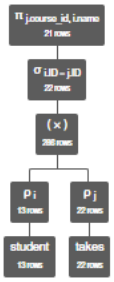
course.course_id	course.title	course.dept_name	course.credits	prereq.prereq_id
'CS-190'	'Game Design'	'Comp. Sci.'	4	'CS-101'
'CS-347'	'Database System Concepts'	'Comp. Sci.'	3	'CS-101'

Q.8) Find the mapping of courses taken and the student name.

```
π j.course_id,i.name ( σ i.ID = j.ID ( ( ρ i ( student) ) × ( ρ j (takes) ) ) )
```

execute query

download history



$$\pi_{\text{j.course\_id, i.name}} (\sigma_{\text{i.ID = j.ID}} ((\rho_{\text{i}} (\text{student})) \times (\rho_{\text{j}} (\text{takes}))))$$

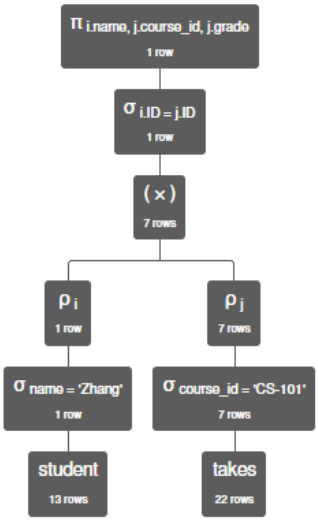
j.course_id	i.name
'CS-101'	'Zhang'
'CS-347'	'Zhang'
'CS-101'	'Shankar'
'CS-190'	'Shankar'
'CS-315'	'Shankar'
'CS-347'	'Shankar'
'HIS-351'	'Brandt'
'FIN-201'	'Chavez'
'PHY-101'	'Peltier'
'CS-101'	'Levy'

Q.9) Find the grade of ‘Zhang’ in the course with course ID CS-101.

```
π i.name,j.course_id,j.grade ( σ i.ID = j.ID ( ( ρ i ( σ name = 'Zhang' (student)) ) × ( ρ j ( σ course_id = 'CS-101' (takes)) ) ) )
```

execute query

download history



$$\pi_{\text{i.name, j.course\_id, j.grade}} (\sigma_{\text{i.ID = j.ID}} ((\rho_{\text{i}} (\sigma_{\text{name = 'Zhang'}} (\text{student}))) \times (\rho_{\text{j}} (\sigma_{\text{course\_id = 'CS-101'}} (\text{takes})))))$$

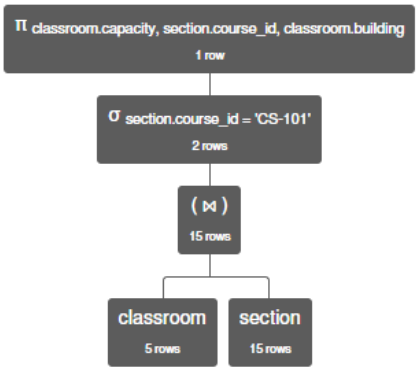
i.name	j.course_id	j.grade
'Zhang'	'CS-101'	'A'

Q.10) Find the room capacity of the course with ID CS-101 running in the spring semester.

```
π σ ρ ← → τ γ ∧ ∨ ¬ = ≠ ≥ ≤ ∩ ∪ ÷ - × ⋈ ⋈ ⋈ ⋈ ⋉ ⋊ ▷ = -- /* {}
1 π classroom.capacity,section.course_id,classroom.building (σ section.course_id='CS-101' ( classroom ⋈ section))
```

execute query

download history



π classroom.capacity, section.course_id, classroom.building (σ section.course_id = 'CS-101' (classroom ⋈ section))

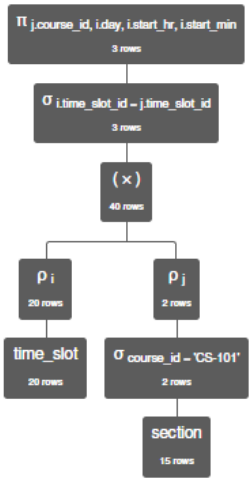
classroom.capacity	section.course_id	classroom.building
500	'CS-101'	'Packard'

Q.11) Find all the time slots (day, start hour, start min) of the couse with course ID CS-101.

```
π j.course_id, i.day, i.start_hr, i.start_min ( σ i.time_slot_id = j.time_slot_id ( ( ρ i ( time_slot) ) × ( ρ j ( σ course_id = 'CS-101' (section) ) ) ) )
```

execute query

download history



$$\pi_{\text{j.course\_id, i.day, i.start\_hr, i.start\_min}} (\sigma_{\text{i.time\_slot\_id = j.time\_slot\_id}} ((\rho_{\text{i}} (\text{time\_slot})) \times (\rho_{\text{j}} (\sigma_{\text{course\_id = 'CS-101'}} (\text{section})))))$$

j.course_id	i.day	i.start_hr	i.start_min
'CS-101'	'T'	14	30
'CS-101'	'R'	14	30
'CS-101'	'W'	10	0