資工三 109590004 呂育瑋

Q1(a).

Candidate key : { A, B, D }

{ A, B } → { C }, ABD → ABCD (Augmentation and reflexivity)

{ B, D } → { E, F }, ABD → ABDEF (Augmentation and reflexivity)

{ A, D } → { G, H }, ABD → ABDGH (Augmentation and reflexivity)

{ A } → { I }, ABD → ABDI (Augmentation and reflexivity)

{ H } → { J }, because { A, D } → { G, H } and { H } → { J } hence ABD → ABDJ (Transitivity)

Union all of them : ABD → ABCDEFGHIJ

Thus, the key of R = { A, B, D }

Q1(b).

R_1 = { A, B, C }

R_2 = { B, D, E, F }

R_3 = { A, D, G, H, J } (merge { A, D } → { G, H } and { H } → { J })

R_4 = { A, I }

Q1(c).

R_1 = { A, B, C }

R_2 = { B, D, E, F }

$R_{3\text{-}1}$ = { A, D, G, H }

$R_{3\text{-}2}$ = { H, J }

R_4 = { A, I }

Q2(a).

There is no multi-value in R, it's in 1NF

Find key first:

Key = { Course_no, Sec_no, Semester, Year } and { Room_no, Days_hours, Semester, Year }

{ CN } → { OD, CH, CL } happened partical dependency

So R is in 1NF.

Q2(b).

(All attribute in short term when calculate)

R = { CN, SN, OD, CH, CL, IS, S, Y, DH, RN, NS }

Candidate key : { CN, SN, S, Y } and { RN, DH, S, Y }

{ CN } → { OD, CH, CL }

{ CN, SN, S, Y } → { DH, RN, NS, IS }

{ RN, DH, S, Y } → { IS, CN, SN }

(1) Check candidate key : { CN, SN, S, Y }

{ CN, SN, S, Y } → { DH, RN, NS, IS } and { CN } → { OD, CH, CL }

Hence { CN, SN, S, Y } → { CN, SN, S, Y, DH, RN, NS, IS, OD, CH, CL } = R

So { CN, SN, S, Y } is key

(2) Check candidate key : { RN, DH, S, Y }

{ RN, DH, S, Y } → { IS, CN, SN } and { CN } → { OD, CH, CL }

Hence { RN, DH, S, Y } → { RN, DH, S, Y, IS, CN, SN, OD, CH, CL }

By { CN, SN, S, Y } → { DH, RN, NS, IS } found NS can depend on { CN, SN, S, Y }

which already in { RN, DH, S, Y }$^{+}$

Hence { RN, DH, S, Y } → { RN, DH, S, Y, IS, CN, SN, OD, CH, CL, NS } = R

So { RN, DH, S, Y } is also a key

Ans: K_1 = { Course_no, Sec_no, Semester, Year }

K_2 = { Room_no, Days_hours, Semester, Year }

Q2(c).

R = { CN, SN, OD, CH, CL, IS, S, Y, DH, RN, NS }

{ CN } → { OD, CH, CL }

{ CN, SN, S, Y } → { DH, RN, NS, IS }

{ RN, DH, S, Y } → { IS, CN, SN }

Remove the transitive dependency. (Separate R)

{ CN } → { OD, CH, CL }

Got R1 = { CN, OD, CH, CL } with Key = { CN }

{ RN, DH, S, Y } → { IS, CN, SN },

{ RN, DH, S, Y } → { RN, DH, S, Y, IS, CN, SN }

Now { RN, DH, S, Y }+ include { CN, SN, S, Y },

{ RN, DH, S, Y } → { RN, DH, S, Y, IS, CN, SN, NS } will keep the FD of { CN, SN, S, Y }

Hence { RN, DH, S, Y } → { RN, DH, S, Y, IS, CN, SN, NS }

Got R2 = { RN, DH, S, Y, IS, CN, SN, NS } with key = { CN, SN, S, Y }

Ans:

R1 = { Course_no, Offering_dept, Credit_hours, Course_level }

FD in R1

{ Course_no } → { Offering_dept, Credit_hours, Course_level }

R2 = {Room_no, Days_hours, Semester, Year, Instructor_ssn, Course_no, Sec_no, No_of_students}

FD in R2

{ Course_no, Sec_no, Semester, Year } → { Days_hours, Room_no, No_of_students, Instructor_ssn }

{ Room_no, Days_hours, Semester, Year } → { Instructor_ssn, Course_no, Sec_no }

Q3(a).

(30 + 9 + 9 + 40 + 10 + 8 + 1 + 4 + 4) + 1 = 116 bytes

Q3(b).

Bfr = floor(512 / 116) = 4 records / block

b = 30000 / 4 = 7500

Q3(c)(i).

R_i = (SSN + P) = (9 + 6) = 15 bytes

bfr_i = floor(B / R_i) = floor(512 / 15) = 34 index records / block

Q3(c)(ii).

r_1 = number of file blocks b = 7500 entries

b_1 = ceiling(r_1 / bfr_i) = ceiling(7500 / 34) = 221 blocks

Q3(c)(iii).

$r_2 = b_1$ = 221 entries

b_2 = ceiling(r_2 / bfr_i) = ceiling(221 / 34) = 7 blocks

$r_3 = b_2$ = 7 entries

b_3 = ceiling(r_3 / bfr_i) = ceiling(7 / 34) = 1

Because b_3 = 1 is the top index level

So the number of levels is 3.

Q3(c)(iv).

$b_i = b_1 + b_2 + b_3$ = 221 + 7 + 1 = 229 blocks

Q3(c)(v).

Number of block accesses to search for a record = the number of levels + 1 = 3 + 1 = 4

Q3(d)(i).

R_i = (SSN + P) = (9 + 6) = 15 bytes

bfr_i = floor(B / R_i) = floor(512 / 15) = 34 index records / block

Q3(d)(ii).

r_1 = number of file records r = 30000 entries

b_1 = ceiling(r_1 / bfr_i) = ceiling(30000 / 34) = 883 blocks

Q3(d)(iii).

$r_2 = b_1$ = 883 entries

b_2 = ceiling(r_2 / bfr_i) = ceiling(883 / 34) = 26 blocks

$r_3 = b_2$ = 26 entries

b_3 = ceiling(r_3 / bfr_i) = ceiling(26 / 34) = 1

Because b_3 = 1 is the top index level

So the number of levels is 3.

Q3(d)(iv).

$b_i = b_1 + b_2 + b_3$ = 883 + 26 + 1 = 910 blocks

Q3(d)(v).

Number of block accesses to search for a record = the number of levels + 1 = 3 + 1 = 4

Q4

Primary Index is on the ordering key field of an ordered file.

Secondary Indexes is on any non-ordering field of a file (ordered, unordered or hashed)

Clustering Indexes is on the ordering non-key field of a file (ordered)