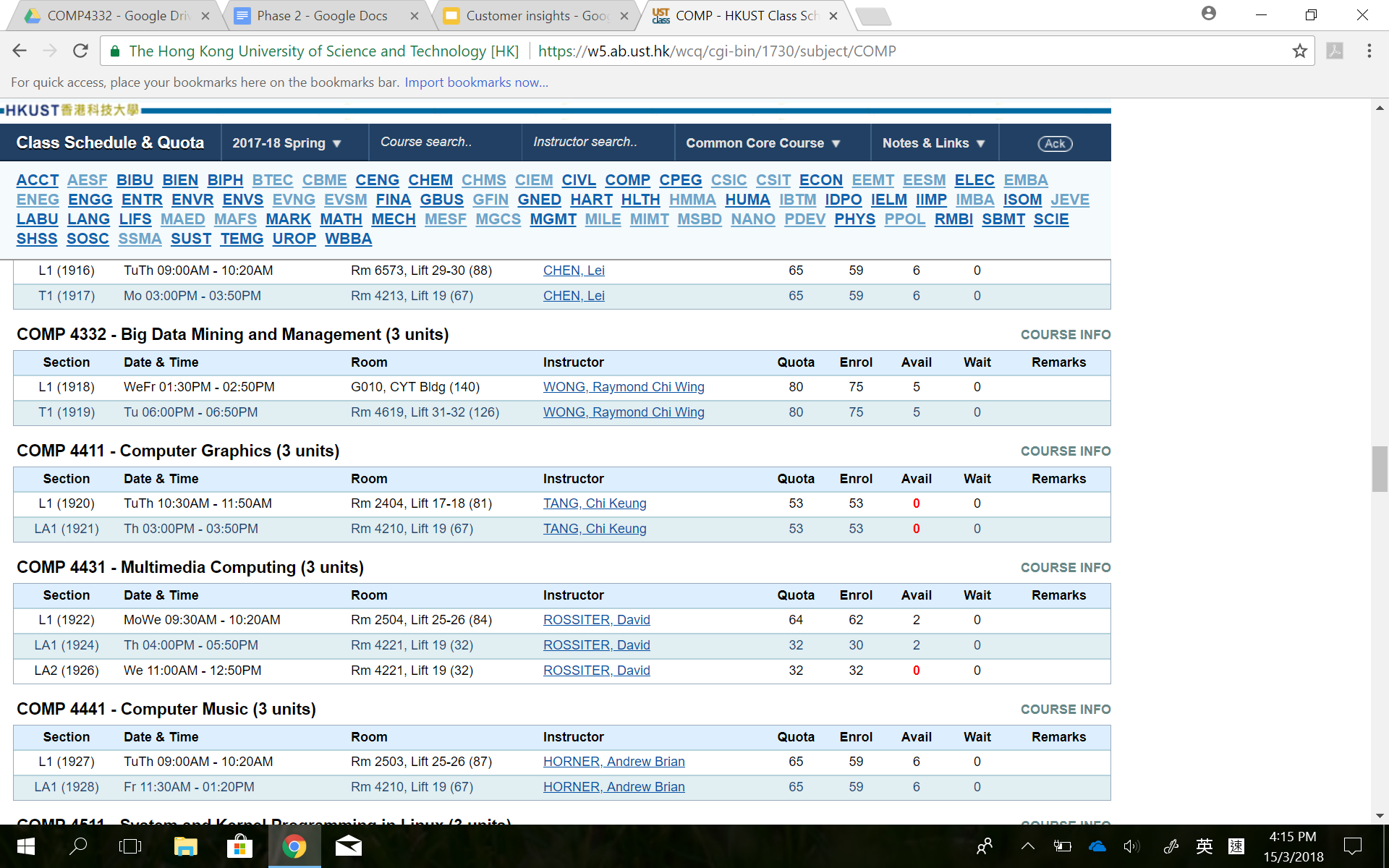
**Database Format**





We categorize the data of a course into course level and section level.

Fields in the course level do not change across time slots. They are further segmented to:

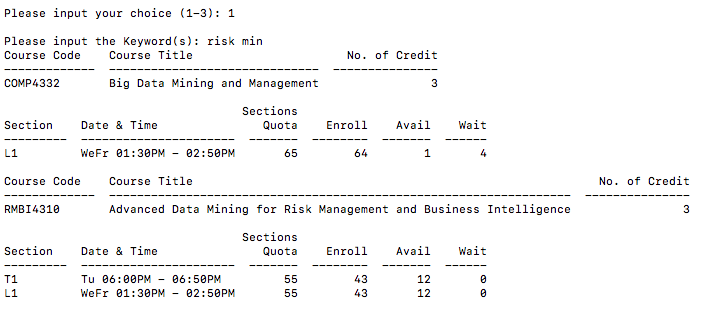
* Mandatory: code, ctitle, credit, description
* Optional: prerequisite, corequisite, colist

Fields in the section level can be changed across time slots. Therefore, the information of time slot is stored in the section level. The fields are further segmented to:

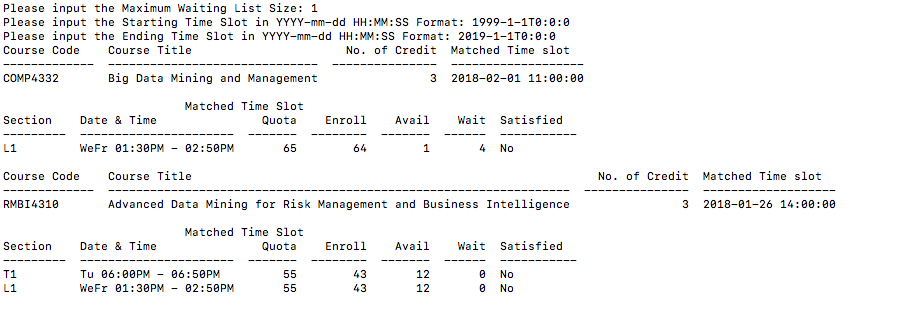
* Mandatory: section, room, instructors, quota, enrol, avail, wait
* Optional: remarks

**Interface:**

**Search By Keywords:**

****

**Search By Size:**

****

**Query in Python**

//5.3.1 Search By Keywords

keywords = input("Please input the Keyword(s): ")

//get the latest datetime

db.course.aggregate([ {"$unwind": "$listOfSections"},\

{"$group": {"\_id":"$code","maxDate":{"$max": "$listOfSections.timeSlot"}}},\

{"$out": "R1"}])

query = []

listOfKeywords = keywords.split(" ")

//append the condition so that the query can handle finding courses that suit one or more keywords

for word in listOfKeywords:

reg =".\*"+word+".\*"

query.append({'ctitle': {'$regex':reg }})

query.append({'description': {'$regex':reg }})

query.append({'listOfSections.remarks': {'$regex':reg }})

//execute the query to get course

listOfCourse=listCourse = db.course.aggregate([{ "$match": { "$or": query } },\

{"$lookup":{ \

"localField": "code",\

"from": "R1",\

"foreignField": "\_id",\

"as": "R3" }},\

{"$unwind": "$R3"},{"$unwind": "$listOfSections"},\

{ "$project": { '\_id': 0, 'code': 1, 'ctitle': 1, 'credit': 1, "listOfSections.section": 1, \

"listOfSections.dateTime": 1, "listOfSections.quota": 1, "listOfSections.enrol": 1,\

"listOfSections.avail": 1, "listOfSections.wait": 1 ,"compareResult": {"$eq": ["$listOfSections.timeSlot", "$R3.maxDate"]}} },\

{"$match": {"compareResult": True}},\

{ "$group": {"\_id": { "code": "$code","ctitle": "$ctitle" ,"credit":"$credit"}, "listOfSections":{"$addToSet": "$listOfSections"}}},\

{"$project":{"code":"$\_id.code","ctitle":"$\_id.ctitle","credit":"$\_id.credit", "listOfSections":"$listOfSections","\_id":0}},\

{ "$sort": { "code": 1 } }])

for course in listOfCourse:

printCourse(course, "")

db.R1.drop()

# 5.3.2 Course Search by Waiting List Size

size = input("Please input the Maximum Waiting List Size: ")

start\_ts = input("Please input the Starting Time Slot in YYYY-mm-ddTHH:MM:SS Format: ")

end\_ts = input("Please input the Ending Time Slot in YYYY-mm-ddTHH:MM:SS Format: ")

//convert input to datetime

dateTime1 = datetime.datetime.strptime(start\_ts, "%Y-%m-%dT%H:%M:%S")

dateTime2 = datetime.datetime.strptime(end\_ts, "%Y-%m-%dT%H:%M:%S")

//get the latest datetime

db.course.aggregate([ {"$unwind": "$listOfSections"},\

{"$group": {"\_id":"$code","maxDate":{"$max": "$listOfSections.timeSlot"}}},\

{"$out": "R1"} ])

Execute query to get list of courses

listOfCourse = db.course.aggregate([\

{"$match": {\

"$and": [{ "listOfSections.timeSlot": { "$gte": dateTime1 } }, \

{ "listOfSections.timeSlot": { "$lte": dateTime2 } }\

]}},\

{"$lookup":{\

"localField": "code",\

"from": "R1",\

"foreignField": "\_id",\

"as": "R3"\

}},\

{"$unwind": "$R3"},\

{"$project":{"\_id": 0, "code":1, "ctitle":1, "credit":1, "listOfSections.section": 1, "listOfSections.dateTime": 1,\

"listOfSections.quota": 1, "listOfSections.enrol": 1, "listOfSections.avail": 1, "listOfSections.wait": 1, "listOfSections.timeSlot": 1, "R3.maxDate":1}},\

{"$unwind": "$listOfSections"},\

{"$project":{"\_id": 0, "code":1, "ctitle":1, "credit":1, "listOfSections.section": 1, "listOfSections.dateTime": 1, \

"listOfSections.quota": 1, "listOfSections.enrol": 1, "listOfSections.avail": 1, "listOfSections.wait": 1, "listOfSections.timeSlot": 1,

"R3.maxDate":1, "compareResult": {"$eq": ["$listOfSections.timeSlot", "$R3.maxDate"]} }},\

{"$match": {"compareResult": True}},\

{"$project":{"\_id": 0, "code":1, "ctitle":1, "credit":1, "listOfSections.section": 1, "listOfSections.dateTime": 1,\

"listOfSections.quota": 1, "listOfSections.enrol": 1, "listOfSections.avail": 1, "listOfSections.wait": 1, "match\_ts":"$R3.maxDate" }},\

{ "$group": {"\_id": { "code": "$code","ctitle": "$ctitle" ,"credit":"$credit" ,"match\_ts":"$match\_ts"}, "listOfSections":{"$addToSet": "$listOfSections"}}},\

{"$project":{"code":"$\_id.code","ctitle":"$\_id.ctitle","credit":"$\_id.credit","match\_ts":"$\_id.match\_ts" ,"listOfSections":"$listOfSections","\_id":0}},\

{ "$sort": { "code": 1 } }\

])

for course in listOfCourse:

printCourse(course,size)

def printSection(sections, size):

//for 5.3.1

if size == "":

table=[]

for section in sections:

temp=[section["section"], section["dateTime"], section["quota"], section["enrol"], section["avail"], section["wait"]]

table.append(temp)

print (tabulate(table, headers={"Section":"", "Date & Time":"", "Quota":"","Enroll":"","Avail":"","Wait":""}))

print("")

//for 5.3.2 , if wait bigger or equal than enrol times size, append “Satisfied” to “Yes” else quals to “No”

else:

table=[]

for section in sections:

wait = float(section["wait"])

enrol = float(section["enrol"])

size = float(size)

condition= wait >= (section["enrol"]\*size)

if(condition):

temp=[section["section"], section["dateTime"], section["quota"], section["enrol"], section["avail"], section["wait"],"Yes"]

table.append(temp)

else:

temp=[section["section"], section["dateTime"], section["quota"], section["enrol"], section["avail"], section["wait"],"No"]

table.append(temp)

print (tabulate(table, headers={"Section":"", "Date & Time":"", "Quota":"","Enroll":"","Avail":"","Wait":"","Satisfied":""}))

print("")

**Insertion**

//The insertion of each course is done by a single script. The “course” collection will contain the course code, course title, description , etc to help filtering the courses by keyword that matches course title, description or remarks in requirement 5.3.1.

ListOfSection inside “course” is an array containing all the sections belongs to the course code and each of the section will have its section number, timeslot ,quota ,etc.

db.course.insert({

code: "COMP4332",

ctitle: "Big Data Mining and Management",

credit: "3",

prerequisite: [{ code: "COMP4211" }, { code: "COMP4331" }, { code: "ISOM3360" }],

colist: [{ code: "RMBI4310" }],

description: "This course will expose students to new and practical issues of real world mining and managing big data. Data mining and management is to effectively support storage, retrieval, and extracting implicit, previously unknown, and potentially useful knowledge from data. This course will place emphasis on two parts. The first part is big data issues such as mining and managing on distributed data, sampling on big data and using some cloud computing techniques on big data. The second part is applications of the techniques learnt on areas such as business intelligence, science and engineering, which aims to uncover facts and patterns in large volumes of data for decision support. This course builds on basic knowledge gained in the introductory data-mining course, and explores how to more effectively mine and manage large volumes of real-world data and to tap into large quantities of data. Working on real world data sets, students will experience all steps of a data-mining and management project, beginning with problem definition and data selection, and continuing through data management, data exploration, data transformation, sampling, portioning, modeling, and assessment.",

listOfSections: [{

section: "L1",

dateTime: "WeFr 01:30PM - 02:50PM",

room: "G010, CYT Bldg (140)",

instructor: "WONG, Raymond Chi Wing",

quota: 65,

enrol: 64,

avail: 1,

wait: 4,

timeSlot: new Date("2018-01-26T14:00:00Z")

},

{

section: "T1",

dateTime: "Tu 06:00PM - 06:50PM",

room: "Rm 4619, Lift 31-32 (126)",

instructor: "WONG, Raymond Chi Wing",

quota: 65,

enrol: 64,

avail: 1,

wait: 4,

timeSlot: new Date("2018-01-26T14:00:00Z")

},

{

section: "L1",

dateTime: "WeFr 01:30PM - 02:50PM",

room: "G010, CYT Bldg (140)",

instructor: "WONG, Raymond Chi Wing",

quota: 65,

enrol: 64,

avail: 1,

wait: 4,

timeSlot: new Date("2018-02-01T11:00:00Z")

}

]

})

**Query**

//find the lastest time slot and it is used by 5.3.1 and 5.3.2

db.course.aggregate([

{$unwind: "$listOfSections"},

{$group: {\_id:"$code",maxDate:{$max: "$listOfSections.timeSlot"}}},

{$out: "R1"}

])

//5.3.1 Course Search by Keyword

//Keyword: "Risk Mining"

//Sorting in ascending order of "Sections" within a single course in Python

db.course.aggregate([

{ $match: { $or: [{ ctitle: /.\*Risk.\*/ }, { description: /.\*Risk.\*/ }, { "listOfSections.remarks": /.\*Risk.\*/ }, { ctitle: /.\*Mining.\*/ },

{ description: /.\*Mining.\*/ }, { "listOfSections.remarks": /.\*Mining.\*/ }] } },

{$lookup:{

localField: "code",

from: "R1",

foreignField: "\_id",

as: "R3"}},

{$unwind: "$R3"},

{$unwind: "$listOfSections"},

{ $project: { \_id: 0, code: 1, ctitle: 1, credit: 1, "listOfSections.section": 1, "listOfSections.dateTime": 1, "listOfSections.quota": 1, "listOfSections.enrol": 1, "listOfSections.avail": 1, "listOfSections.wait": 1 ,compareResult: {$eq: ["$listOfSections.timeSlot", "$R3.maxDate"]} } },

{$match: {compareResult: true}},

{ $group: {\_id: { "code": "$code","ctitle": "$ctitle" ,"credit":"$credit"}, "listOfSections":{"$addToSet": "$listOfSections"}}},

{$project:{code:"$\_id.code",ctitle:"$\_id.ctitle",credit:"$\_id.credit", listOfSections:"$listOfSections",\_id:0}},

{ $sort: { code: 1 } }])

//To implement the above function, we will first split the the keyword string into several words (do in pthon). Then, we will check every keyword whether it matches the title , description or remarks of the course by using $match and regular expressions. At last, we output the result according to the project specification, such as corse title, credit , and its list of sections and sort them in ascending order.

The expected result:

{ "code" : "COMP4332", "ctitle" : "Big Data Mining and Management", "credit" : "3", "listOfSections" : [ { "section" : "L1", "dateTime" : "WeFr 01:30PM - 02:50PM", "quota" : 65, "enrol" : 64, "avail" : 1, "wait" : 4 } ] }

{ "code" : "RMBI4310", "ctitle" : "Advanced Data Mining for Risk Management and Business Intelligence", "credit" : "3", "listOfSections" : [ { "section" : "T1", "dateTime" : "Tu 06:00PM - 06:50PM", "quota" : 55, "enrol" : 43, "avail" : 12, "wait" : 0 }, { "section" : "L1", "dateTime" : "WeFr 01:30PM - 02:50PM", "quota" : 55, "enrol" : 43, "avail" : 12, "wait" : 0 } ] }

//5.3.2 Course Search by Waiting list size

//f: 0.05, start\_ts: 2018-01-26T14:00:00Z, end\_ts: 2018-02-01T11:30:00Z

db.course.aggregate([

//join the table with R1

{$lookup:{

localField: "code",

from: "R1",

foreignField: "\_id",

as: "R3"}},

{$unwind: "$R3"},

{$project:{\_id: 0, code:1, ctitle:1, credit:1, "listOfSections.section": 1, "listOfSections.dateTime": 1, "listOfSections.quota": 1, "listOfSections.enrol": 1, "listOfSections.avail": 1, "listOfSections.wait": 1, "listOfSections.timeSlot": 1, "R3.maxDate":1}},

{$unwind: "$listOfSections"},

//find the sections with time slot = maxDate

{$project:{\_id: 0, code:1, ctitle:1, credit:1, "listOfSections.section": 1, "listOfSections.dateTime": 1, "listOfSections.quota": 1, "listOfSections.enrol": 1, "listOfSections.avail": 1, "listOfSections.wait": 1, "listOfSections.timeSlot": 1, "R3.maxDate":1, compareResult: {$eq: ["$listOfSections.timeSlot", "$R3.maxDate"]} }},

{$match: {compareResult: true}},

//output the information

{$project:{\_id: 0, code:1, ctitle:1, credit:1, "listOfSections.section": 1, "listOfSections.dateTime": 1, "listOfSections.quota": 1, "listOfSections.enrol": 1, "listOfSections.avail": 1, "listOfSections.wait": 1, match\_ts:"$R3.maxDate" }},

{ $group: {\_id: { "code": "$code","ctitle": "$ctitle" ,"credit":"$credit" ,"match\_ts":"$match\_ts"}, "listOfSections":{"$addToSet": "$listOfSections"}}},

{$project:{code:"$\_id.code",ctitle:"$\_id.ctitle",credit:"$\_id.credit", match\_ts:"$\_id.ctitle" ,listOfSections:"$listOfSections",\_id:0}},

{ $sort: { "\_id.code": 1 } }

])

db.R1.drop()

The above function is first join the table created in the last function to get the latest timeslot of each course, then it will check whether the course timeslot matches the latest timeslot with compareResult: {$eq: ["$listOfSections.timeSlot", "$R2.maxDate"]} }}. After output the result and it in ascending order in section, we drop the temporary table created in the last fiction to release spaces. For the attribute “Satisfied” , we will use the number of enrol and wait to calculate inside python.

The expected result :

{ "code" : "RMBI4310", "ctitle" : "Advanced Data Mining for Risk Management and Business Intelligence", "credit" : "3", "match\_ts" : "Advanced Data Mining for Risk Management and Business Intelligence", "listOfSections" : [ { "section" : "T1", "dateTime" : "Tu 06:00PM - 06:50PM", "quota" : 55, "enrol" : 43, "avail" : 12, "wait" : 0 }, { "section" : "L1", "dateTime" : "WeFr 01:30PM - 02:50PM", "quota" : 55, "enrol" : 43, "avail" : 12, "wait" : 0 } ] }

{ "code" : "COMP4332", "ctitle" : "Big Data Mining and Management", "credit" : "3", "match\_ts" : "Big Data Mining and Management", "listOfSections" : [ { "section" : "L1", "dateTime" : "WeFr 01:30PM - 02:50PM", "quota" : 65, "enrol" : 64, "avail" : 1, "wait" : 4 } ] }