Module Test SQL

**select** count(\*) **from** cd.facilities;

**select** count(\*) **from** cd.facilities **where** guestcost >= 10;

**select** recommendedby, count(\*) **from** cd.members **where** recommendedby **is** **not** **null** **group** **by** recommendedby **order** **by** recommendedby;

**select** facid, sum(slots) **as** "Total Slots" **from** cd.bookings **group** **by** facid **order** **by** facid;

**select** facid, sum(slots) **as** "Total Slots" **from** cd.bookings **where** starttime >= '2012-09-01' **and** starttime < '2012-10-01' **group** **by** facid **order** **by** sum(slots);

**select** facid, extract(month **from** starttime) **as** month, sum(slots) **as** "Total Slots" **from** cd.bookings **where** starttime >= '2012-01-01' **and** starttime < '2013-01-01' **group** **by** facid, month **order** **by** facid, month;

**select** count(**distinct** memid) **from** cd.bookings

**select** facid, sum(slots) **as** "Total Slots" **from** cd.bookings **group** **by** facid **having** sum(slots) > 1000 **order** **by** facid

**select** facs.name, sum(slots \* **case** **when** memid = 0 **then** facs.guestcost **else** facs.membercost **end**) **as** revenue **from** cd.bookings bks **inner** **join** cd.facilities facs **on** bks.facid = facs.facid **group** **by** facs.name **order** **by** revenue;

**select** name, revenue **from** ( **select** facs.name, sum(**case** **when** memid = 0 **then** slots \* facs.guestcost **else** slots \* membercost **end**) **as** revenue **from** cd.bookings bks **inner** **join** cd.facilities facs **on** bks.facid = facs.facid **group** **by** facs.name ) **as** agg **where** revenue < 1000 **order** **by** revenue;

**select** facid, sum(slots) **as** "Total Slots" **from** cd.bookings **group** **by** facid **order** **by** sum(slots) **desc** LIMIT 1;

**select** facid, extract(month **from** starttime) **as** month, sum(slots) **as** slots **from** cd.bookings **where** starttime >= '2012-01-01' **and** starttime < '2013-01-01' **group** **by** rollup(facid, month) **order** **by** facid, month

**select** facs.facid, facs.name, trim(to\_char(sum(bks.slots)/2.0, '9999999999999999D99')) **as** "Total Hours" **from** cd.bookings bks **inner** **join** cd.facilities facs **on** facs.facid = bks.facid **group** **by** facs.facid, facs.name **order** **by** facs.facid;

**select** mems.surname, mems.firstname, mems.memid, min(bks.starttime) **as** starttime **from** cd.bookings bks **inner** **join** cd.members mems **on** mems.memid = bks.memid **where** starttime >= '2012-09-01' **group** **by** mems.surname, mems.firstname, mems.memid **order** **by** mems.memid;

**select** count(\*) **over**(), firstname, surname **from** cd.members **order** **by** joindate

**select** row\_number() **over**(**order** **by** joindate), firstname, surname **from** cd.members **order** **by** joindate

**select** facid, total **from** ( **select** facid, sum(slots) total, rank() **over** (**order** **by** sum(slots) **desc**) rank **from** cd.bookings **group** **by** facid ) **as** ranked **where** rank = 1

**select** firstname, surname, ((sum(bks.slots)+10)/20)\*10 **as** hours, rank() **over** (**order** **by** ((sum(bks.slots)+10)/20)\*10 **desc**) **as** rank **from** cd.bookings bks **inner** **join** cd.members mems **on** bks.memid = mems.memid **group** **by** mems.memid **order** **by** rank, surname, firstname;

**select** name, rank **from** ( **select** facs.name **as** name, rank() **over** (**order** **by** sum(**case** **when** memid = 0 **then** slots \* facs.guestcost **else** slots \* membercost **end**) **desc**) **as** rank **from** cd.bookings bks **inner** **join** cd.facilities facs **on** bks.facid = facs.facid **group** **by** facs.name ) **as** subq **where** rank <= 3 **order** **by** rank;

**select** name, **case** **when** class=1 **then** 'high' **when** class=2 **then** 'average' **else** 'low' **end** revenue **from** ( **select** facs.name **as** name, ntile(3) **over** (**order** **by** sum(**case** **when** memid = 0 **then** slots \* facs.guestcost **else** slots \* membercost **end**) **desc**) **as** class **from** cd.bookings bks **inner** **join** cd.facilities facs **on** bks.facid = facs.facid **group** **by** facs.name ) **as** subq **order** **by** class, name;

**select** facs.name **as** name, facs.initialoutlay/((sum(**case** **when** memid = 0 **then** slots \* facs.guestcost **else** slots \* membercost **end**)/3) - facs.monthlymaintenance) **as** months **from** cd.bookings bks **inner** **join** cd.facilities facs **on** bks.facid = facs.facid **group** **by** facs.facid **order** **by** name;

**select** dategen.date, ( *-- correlated subquery that, for each day fed into it,* *-- finds the average revenue for the last 15 days* **select** sum(**case** **when** memid = 0 **then** slots \* facs.guestcost **else** slots \* membercost **end**) **as** rev **from** cd.bookings bks **inner** **join** cd.facilities facs **on** bks.facid = facs.facid **where** bks.starttime > dategen.date - interval '14 days' **and** bks.starttime < dategen.date + interval '1 day' )/15 **as** revenue **from** ( *-- generates a list of days in august* **select** cast(generate\_series(timestamp '2012-08-01', '2012-08-31','1 day') **as** date) **as** date ) **as** dategen **order** **by** dategen.date;

Timestamps

**select** timestamp '2012-08-31 01:00:00';

**select** timestamp '2012-08-31 01:00:00' - timestamp '2012-07-30 01:00:00' **as** interval

**select** generate\_series(timestamp '2012-10-01', timestamp '2012-10-31', interval '1 day') **as** ts

**select** extract(day **from** timestamp '2012-08-31');

**select** extract(epoch **from** (timestamp '2012-09-02 00:00:00' - '2012-08-31 01:00:00'));

**select** extract(month **from** cal.month) **as** month, (cal.month + interval '1 month') - cal.month **as** length **from** ( **select** generate\_series(timestamp '2012-01-01', timestamp '2012-12-01', interval '1 month') **as** month ) cal **order** **by** month;

**select** (date\_trunc('month',ts.testts) + interval '1 month') - date\_trunc('day', ts.testts) **as** remaining **from** (**select** timestamp '2012-02-11 01:00:00' **as** testts) ts

**select** starttime, starttime + slots\*(interval '30 minutes') endtime **from** cd.bookings **order** **by** endtime **desc**, starttime **desc** limit 10

**select** date\_trunc('month', starttime) **as** month, count(\*) **from** cd.bookings **group** **by** month **order** **by** month

**select** name, month, round((100\*slots)/ cast( 25\*(cast((month + interval '1 month') **as** date) - cast (month **as** date)) **as** numeric),1) **as** utilisation **from** ( **select** facs.name **as** name, date\_trunc('month', starttime) **as** month, sum(slots) **as** slots **from** cd.bookings bks **inner** **join** cd.facilities facs **on** bks.facid = facs.facid **group** **by** facs.facid, month ) **as** inn **order** **by** name, month

Operations

**select** surname || ', ' || firstname **as** name **from** cd.members

**select** \* **from** cd.facilities **where** name **like** 'Tennis%';

**select** \* **from** cd.facilities **where** upper(name) **like** 'TENNIS%';

**select** memid, telephone **from** cd.members **where** telephone ~ '[()]';

**select** lpad(cast(zipcode **as** char(5)),5,'0') zip **from** cd.members **order** **by** zip

**select** substr (mems.surname,1,1) **as** letter, count(\*) **as** count **from** cd.members mems **group** **by** letter **order** **by** letter

**select** memid, translate(telephone, '-() ', '') **as** telephone **from** cd.members **order** **by** memid;

Queries

**with** recursive recommenders(recommender) **as** ( **select** recommendedby **from** cd.members **where** memid = 27 **union** **all** **select** mems.recommendedby **from** recommenders recs **inner** **join** cd.members mems **on** mems.memid = recs.recommender ) **select** recs.recommender, mems.firstname, mems.surname **from** recommenders recs **inner** **join** cd.members mems **on** recs.recommender = mems.memid **order** **by** memid **desc**

**with** recursive recommendeds(memid) **as** ( **select** memid **from** cd.members **where** recommendedby = 1 **union** **all** **select** mems.memid **from** recommendeds recs **inner** **join** cd.members mems **on** mems.recommendedby = recs.memid ) **select** recs.memid, mems.firstname, mems.surname **from** recommendeds recs **inner** **join** cd.members mems **on** recs.memid = mems.memid **order** **by** memid

**with** recursive recommenders(recommender, member) **as** ( **select** recommendedby, memid **from** cd.members **union** **all** **select** mems.recommendedby, recs.member **from** recommenders recs **inner** **join** cd.members mems **on** mems.memid = recs.recommender ) **select** recs.member member, recs.recommender, mems.firstname, mems.surname **from** recommenders recs **inner** **join** cd.members mems **on** recs.recommender = mems.memid **where** recs.member = 22 **or** recs.member = 12 **order** **by** recs.member **asc**, recs.recommender **desc**