```sql
/*
1.  Create a dashboard which depicts the number of students who are
    currently studying and dropped out of university from each major.
2.  Who should be invited to an Alumni Happy Hour in Texas?
3.  Who should receive the most recent copy of INAG News?
4.  What type of the listservs are the alumni, prospectives, and students
    currently enrolled right now?
5.  What is the latest communication date of each event type?
6.  How many communications were sent for each event. How many have been
    received and how many are yet to be sent?
*/

USE students_communication;

/* Question 1 */

DROP VIEW IF EXISTS student_summary_view;

CREATE VIEW student_summary_view AS
WITH ongoing AS (
      SELECT major
                 , COUNT(student_id) as ongoing_std
      FROM students
      WHERE status != 'DropOut'
      GROUP BY major
),

dropout as (
      SELECT major
                 , COUNT(student_id) as dropouts
                 , COUNT(CASE WHEN description = 'Personal' THEN
student_id END) AS personal_drp
                 , COUNT(CASE WHEN description = 'Financial' THEN
student_id END) AS financial_drp
                 , COUNT(CASE WHEN description = 'Academic' THEN
student_id END) AS academic_drp
      FROM students
      WHERE status = 'DropOut'
      GROUP BY major
),

totals as (
      SELECT major
                 , COUNT(student_id) as total_students
      FROM students
      GROUP BY major
)

SELECT t.major as "Major"
      , total_students as "Total Students"
    , ongoing_std as "Ongoing Students"
    , dropouts as "Dropout Students"
    , personal_drp as "Personal Dropouts"
    , financial_drp as "Financial Dropouts"
    , academic_drp as "Academic Dropouts"
FROM totals t
LEFT JOIN ongoing o ON t.major = o.major
LEFT JOIN dropout d ON t.major = d.major
ORDER BY total_students DESC, ongoing_std DESC;

SELECT * FROM student_summary_view;

/* Question 2 */
DELIMITER //

DROP PROCEDURE IF EXISTS GetAlumniByState //

CREATE PROCEDURE GetAlumniByState(IN state_name VARCHAR(45))
BEGIN
    SELECT DISTINCT first_name as "First Name"
    , last_name as "Last Name"
    , personal_email as "Personal Email"
    , address_line_1 as "Address Line 1"
    , address_line_2 as "Address Line 2"
    , city as "City"
    , state as "State"
    , country as "Country"
    , zip as "Zip Code"
    FROM people p
    JOIN alumni al ON p.id = al.id
    LEFT JOIN addresses a ON p.address_id = a.address_id
    WHERE a.good_address = 1 AND a.state = state_name;
END //

DELIMITER ;

CALL GetAlumniByState('Texas');

/* Question 3 */

DROP VIEW IF EXISTS INAG_address_details;

CREATE VIEW INAG_address_details AS
WITH student_details AS (
    SELECT
        p.first_name,
        p.last_name,
        a.  address_line_1,
        a.  address_line_2,
        a.  city,
        a.  state,
        a.  country,
        a.  zip
    FROM people p
    JOIN students s ON s.id = p.id
    LEFT JOIN addresses a ON a.address_id = s.current_address_id
    WHERE s.status != 'DropOut' AND a.good_address = 1
),
person_details AS (
    SELECT
        p.first_name,
        p.last_name,
        a.  address_line_1,
        a.  address_line_2,
        a.  city,
        a.  state,
        a.  country,
        a.  zip
    FROM people p
    LEFT JOIN addresses a ON a.address_id = p.address_id
    WHERE p.status IN ('Alumni','Prospective Student') AND a.good_address
= 1
)
SELECT
    *
FROM (
    SELECT first_name, last_name, address_line_1, address_line_2, city,
state, country, zip
    FROM student_details sd
    UNION ALL
    SELECT first_name, last_name, address_line_1, address_line_2, city,
state, country, zip
    FROM person_details pd
) AS det
ORDER BY first_name ASC;

SELECT * FROM INAG_address_details;

/* Question 4 */

DROP VIEW IF EXISTS enrollment_summary_view;

CREATE VIEW enrollment_summary_view AS
SELECT listserv_description
    , COUNT(CASE WHEN status = 'Prospective Student' THEN p.id END) AS
"Prospective Students Enrolled"
    , COUNT(CASE WHEN status = 'Student' THEN p.id END) AS "Students
Enrolled"
    , COUNT(CASE WHEN status = 'Alumni' THEN p.id END) AS "Alumni
Enrolled"
    , COUNT(CASE WHEN status = 'Dropout' THEN p.id END) AS "Dropouts
Enrolled"
FROM listservs l
JOIN people_listservs pl ON l.listserv_id = pl.listserv_id
JOIN people p ON p.id = pl.people_id
GROUP BY listserv_description WITH ROLLUP
ORDER BY listserv_description DESC;

SELECT * FROM enrollment_summary_view;

/* Question 5 */
```

```
DROP VIEW IF EXISTS event_lastest_communication;

CREATE VIEW event_lastest_communication AS
SELECT event_description AS "Event Description"
      , MAX(comm_date) AS "Latest Communication Time"
FROM events e
LEFT JOIN communications c ON e.event_id = c.event_id
GROUP BY event_description
ORDER BY MAX(comm_date) DESC;

SELECT * FROM event_lastest_communication;

/* Question 6 */

DROP VIEW IF EXISTS event_communications_view;

CREATE VIEW event_communications_view AS
SELECT event_description AS "Event Description",
       COUNT(comm_id) AS "Total Communications Planned",
       COUNT(CASE WHEN comm_status = 'Sent' THEN comm_id END) AS
"Communications Sent",
       COUNT(CASE WHEN comm_status = 'Not Sent' THEN comm_id END) AS
"Communications Not Sent",
       COUNT(CASE WHEN comm_status = 'Received' THEN comm_id END) AS
"Communications Received"
FROM events e
LEFT JOIN communications c ON e.event_id = c.event_id
GROUP BY event_description;

SELECT * FROM event_communications_view;
```